Securing Government Classified and Sensitive Information

Zero Trust, Post-Quantum, Air Gap and Cybersecurity Frameworks

Paul George Savluc

PAUL SAVLUC

Securing Government Classified and Sensitive Information: Zero Trust, Post-Quantum, and Cybersecurity Frameworks

Copyright © 2025 by Paul Savluc

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without written permission from the publisher. It is illegal to copy this book, post it to a website, or distribute it by any other means without permission.

First edition

This book was professionally typeset on Reedsy.
Find out more at reedsy.com

To the public servants, engineers, analysts, and guardians of trust who do the quiet work that keeps a nation safe.
To the mentors who taught me that security is a discipline of humility, and to the builders who prove every day that rigor and integrity can coexist with speed.
And to my family and friends, who patiently live beside the mission.
Kind regards,
Paul George Savluc

"Trust is not a credential. It is a hypothesis that must be tested continuously."
"Security is not a product you buy. It is a system you build, verify, and keep building."

Kind regards,
Paul George Savluc

Contents

Foreword v
Preface vii
Acknowledgments ix
Introduction 1
1 Understanding Information Classification and Sensitivity 3
2 The Cybersecurity Policy Landscape 7
3 Frameworks for Security and Compliance 13
4 Zero Trust Architecture in Government 20
5 Implementing Zero Trust: Federal Initiatives and Components 24
6 Challenges and Considerations: 28
7 Post-Quantum Cryptography and Future Encryption 30
The Quantum Threat 31
8 NIST's Post-Quantum Standards 33
9 Government Transition Plans (NSM-10 and Beyond) 36
10 Secure Systems Engineering and Architecture 40
Secure Configuration Baselines 41
11 Network Architecture and Segmentation 43
12 Hardware Security and Peripheral Controls 46
13 How to Air-Gap Systems for TEMPEST, Signal Security, and the... 49

1. What TEMPEST actually means 50
2. Threat model: who can exploit signal leakage? 50
3. The physics in one page 52
3.1 Coupling paths 52
3.2 Why “digital” still leaks like analog 52
4. What leaks in real systems 53
4.1 Displays and video links 53
4.2 Keyboards and input devices 53
4.3 CPUs, GPUs, memory, and switching regulators 54
4.4 Networking and common-mode emissions 54
5. Signal security is bigger than TEMPEST 55
6. Zoning and RED BLACK separation 55
7. Shielding, bonding, grounding: the holy trinity (and how it goes wrong) 56
8. Filtering and isolation controls 57
8.1 Power line filtering 57
8.2 Data line filtering 57
8.3 Physical isolation 58
9. Operational controls: how organizations fail in practice 58
10. Testing and verification: you cannot secure what you do not measure 59
10.1 A defensive RF survey workflow (high-level) 59
11. Example: a simple defensive spectrum survey script (Python) 60
12. Design patterns that drastically reduce risk 64

13. Implications for modern "secure compute" and AI workloads 65
14. Policy and compliance angle (practical, not mystical) 65
15. A pragmatic checklist 66
Closing thought 67
14 Secure Development and Local Applications 68
15 Monitoring and Incident Response Preparedness 70
16 Secure Communication and Collaboration 73
Encrypted Voice and Video Communications 73
Secure Email and Messaging 75
Inter-Agency and Coalition Communication 77
Human Factors and Operational Security (OPSEC) 79
17 Hardware and Supply Chain Security 82
Hardware Supply Chain 82
18 Software Supply Chain and Third-Party Components 85
19 Monitoring and Responding to Supply Chain Threats 88
20 Physical Supply Chain & Counterfeit Prevention 90
21 Personnel Security and Insider Threats 92
The Security Clearance System 92
Training and Awareness 95
Insider Threat Programs and Monitoring 96
Physical Security & Personnel 98
22 Applications That Never Shut Up 101
Why Software Constantly Leaks Data and Why Secure Systems Must Be Ruthless 101

1. The uncomfortable truth: most applications are not designed to be silent 102
2. Leakage is not only about content 103
3. Backdoors are not always malicious 104
4. The false comfort of "closed source" and "open source" 104
5. The background network problem 105
6. Why "I was not using it" is irrelevant 106
7. Secure systems require software minimalism 106
8. Trust is not binary, it is contextual 107
9. The myth of "offline mode" 108
10. Enforcing silence: architectural controls 108
11. Supply chain and update risks 109
12. Psychological traps that break secure discipline 109
13. Secure systems must assume software is guilty until proven silent 110
14. The core rule 111
Closing thought 111
23 Continuous Monitoring, Incident Response, and Improvement 112
Continuous Monitoring and Detection 112
24 Incident Response (IR) 116
25 Continuous Improvement and Adaptation 119
26 Conclusion 122

Foreword

There are books that describe cybersecurity, and there are books that treat it like engineering, the kind that has to survive weather, time, malice, and human error. This is the latter.

The modern government environment is a paradox you have to learn to enjoy. The mission demands speed, but the stakes demand restraint. The work is distributed, but the secrets must be contained. The tools evolve rapidly, but the adversary evolves faster. In that landscape, security is not a checklist, not a box, not a vibe. Security is a set of decisions made under uncertainty, validated by evidence, and improved when reality disagrees with your assumptions.

The most important shift of the last decade is philosophical: we no longer treat the network as a safe place. We treat it as a contested place. Zero Trust is the formal expression of that realism. It is the operationalization of a simple idea: verification beats hope. Meanwhile, the coming cryptographic transition to post-quantum standards reminds us that time itself is an attacker. Messages intercepted today can be broken tomorrow. The timeline is unknown, but the obligation is not.

This book does not ask you to memorize doctrine. It asks you to build systems that remain dependable under pressure, audit, and attack. It asks you to engineer for correctness, failure, recovery, and accountability, and to do so in a way that aligns with the frameworks the U.S. Government requires and the

scientific standards that reality enforces.

If you work in this space, you already know the truth: the job is bigger than any single tool, vendor, or team. It is a long game played in quarters and milliseconds. The only way to win is to keep learning, keep measuring, and keep reducing the distance between what you believe is true and what your telemetry proves is true.

This book is a field manual for that mindset.

Preface

This book was written for people who have to make security real.

Not hypothetical. Not "in principle." Real, as in: the system has a mission, a budget, a deadline, a human operator, and an adversary who gets a vote.

Securing classified, confidential, and sensitive work is a rare kind of engineering challenge because it spans layers that are usually separated. It is architecture and policy. Cryptography and human behavior. Physical access and endpoint integrity. Procurement and incident response. Documentation and operational discipline. You cannot solve it by excelling at only one layer. The system fails at its seams.

The goal here is to provide an end-to-end engineering approach to government-grade security:

- How to think in risk, not superstition.
- How to implement Zero Trust so it's more than a buzzword.
- How to design local secure environments that can withstand compromise attempts and audits.
- How to build secure communications and hardware connections that do not collapse under real-world constraints.
- How to adopt post-quantum cryptography deliberately, without breaking everything you depend on.
- How to integrate compliance frameworks into engineering

workflows so security becomes repeatable, testable, and provable.

You will see an insistence on measurable controls, layered defenses, and operational feedback loops. You will also see a bias toward pragmatic engineering: strong defaults, minimal assumptions, cryptographic agility, and monitoring that is worthy of the name.

The end state is not "perfect security." That does not exist. The end state is **bounded risk, continuously reduced**, with systems that can detect, respond, recover, and improve.

If you build systems for government work, you are building in the presence of history. Your decisions will outlive your sprint cycles. This book is my attempt to make those decisions less mysterious, more scientific, and more defensible.

Acknowledgments

No serious security work is done alone.

I want to acknowledge the communities and professionals who keep raising the bar, often without public credit: the federal cybersecurity workforce, the incident responders who show up at the worst hour, the architects who argue for correctness when shortcuts are tempting, and the auditors who demand evidence instead of stories.

I also acknowledge the standards bodies and researchers who turn chaos into shared language: the engineers and scientists behind NIST publications, the practitioners shaping Zero Trust doctrine into deployable architectures, and the cryptographers stress-testing post-quantum algorithms with the kind of skepticism that protects everyone.

To the open-source maintainers, reverse engineers, red teamers, blue teamers, and defenders who keep shipping improvements and finding flaws before the adversary does: thank you for making the ecosystem stronger, one commit and one report at a time.

Finally, to the mentors, colleagues, and partners who have challenged my assumptions and sharpened my thinking: your insistence on clarity and proof shaped this work more than you know.

Any remaining mistakes are mine. The responsibility is always local.

Introduction

Protecting classified, confidential, and sensitive information is paramount for national security and public trust in government operations. Cyber threats – from nation-state hackers to insider leaks – continually evolve in sophistication, targeting the very "vital institutions that underpin the American way of life". In the current threat environment, the government **can no longer depend on conventional perimeter-based defenses** alone. As some Presidents warned, *"incremental improvements will not give us the security we need; instead, the Federal Government needs to make bold changes and significant investments"*. This call to action has driven **bold shifts** in strategy: embracing Zero Trust architecture, preparing for post-quantum encryption, and enforcing rigorous cybersecurity rules and frameworks across federal agencies.

This comprehensive guide (essentially a "book") details how to engineer and maintain secure government systems and communications. We will explore how to handle classified and sensitive data properly, implement a **Zero Trust Architecture** (where no user or device is inherently trusted without verification), adopt **post-quantum cryptographic protections** to safeguard data against future quantum computer threats, and abide by major current cybersecurity frameworks, policies, and regulations. Importantly, securing information is not just about deploying the right technology – it's also about processes and

people. Thus, we will also cover how to establish secure local environments, hardened hardware connections, and a culture of security among personnel to achieve a true "**defense-in-depth**" posture.

By weaving together U.S. government requirements (from laws like FISMA to NIST frameworks and recent Executive Orders) with cutting-edge practices (like Zero Trust and cryptographic agility), this guide provides a roadmap to "**do everything**" necessary to protect sensitive work. The approach is **highly current**, reflecting policies and standards as of 2025, and **scientific**, drawing on established frameworks and emerging research with citations for further reference. In the pages ahead, we will delve into each facet – from classifying information correctly to engineering secure networks and software, to preparing for tomorrow's threats – equipping you with knowledge to build and manage systems that can truly safeguard government data and missions.

1

Understanding Information Classification and Sensitivity

Before implementing technical security measures, it is essential to understand how the government classifies and handles information. Different levels of sensitivity require different protections.

- **Classification Levels:** The U.S. uses three primary classification levels – **Confidential, Secret, and Top Secret**. Each level indicates an increasing degree of potential damage to national security if information is disclosed without authorization. Confidential information, if improperly released, could be expected to cause *"damage"* to national security; Secret information could cause *"serious damage"*; and Top Secret information could cause *"exceptionally grave damage"*. These definitions (from Executive Order 13526) set the threshold for classifying information. For example, military plans might be Secret, while the identity of undercover intelligence agents might be Top Secret due to the grave danger disclosure would pose.

- **Clearances and Need-to-Know:** Individuals must have the appropriate **security clearance** to access classified information at a given level. Holding a clearance means a background investigation has been completed and the person is deemed trustworthy to handle information up to that level. **However, clearance alone is not enough** – the person must also have a **need-to-know** the specific information for their job. For instance, a person with a Secret clearance can access Secret and Confidential information, but not Top Secret, and even then only the Secret information relevant to their duties. This principle limits exposure of information only to those who truly require it, reducing insider threat surfaces.
- **Compartments and Special Access:** In addition to the three basic levels, certain information is further compartmentalized. **Sensitive Compartmented Information (SCI)** refers to intelligence (often from sources/methods) that is segregated into compartments; even someone with a Top Secret clearance needs explicit SCI authorization for that compartment to access it. Similarly, **Special Access Programs (SAPs)** restrict information to specially authorized individuals beyond normal clearance, often with additional safeguards. These regimes are used for extremely sensitive projects (e.g., advanced weapon developments or clandestine operations) and impose need-to-know at the granular level of each program.
- **Controlled Unclassified Information (CUI):** Not all sensitive government information is classified. The category of **CUI** was established to cover information that is unclassified but still sensitive and requires controls on its dissemination. Examples include critical infrastruc-

ture information, law enforcement sensitive data, export-controlled technical data, or personally identifiable information. Previously, agencies used ad-hoc labels like *For Official Use Only (FOUO)* or *Sensitive But Unclassified (SBU)*; CUI unifies these under one framework with defined handling practices. CUI must be protected (e.g., with access controls, encryption in transit, etc.) though it is not classified under EO 13526. A **CUI Registry** governs what subcategories of information fall here and what markings to use. In daily work, an analyst might mark a report as *"CUI//NOFORN"* meaning it's controlled unclassified info, not for release to foreign nationals.

- **Marking and Handling:** All classified documents are marked clearly at the top and bottom with their classification (and compartments if any). Portions of text are portion-marked (e.g., "(S)" for Secret) so readers know what is Secret vs. just Confidential in a document. There are also handling caveats like **NOFORN** (no foreign dissemination) or **ORCON** (originator controlled) that further restrict sharing. Physical handling rules require, for example, that Secret and Top Secret materials be stored in approved safes when not in use, and transmitted by secure methods (Top Secret usually via direct courier or encrypted channels, not by regular email). If you're in a secure facility, you'll see colored cover sheets (blue for Confidential, red for Secret, orange for Top Secret) to signal protection level and prevent shoulder-surfing.
- **Declassification and Downgrading:** Classified information can be downgraded or declassified over time, usually after a set number of years or an event (unless it still meets classification criteria). For example, many historical

WWII or Cold War documents have been declassified. Understanding declassification is part of secure work – e.g., knowing what older information can be discussed openly vs. what remains classified. But by default, one must **assume information is classified at its labeled level until an authorized declassification occurs**.

- **Penalties for Mishandling:** Laws such as the **Espionage Act** and others make mishandling classified information a crime. High-profile cases of leaking or espionage have led to prosecutions. While this book focuses on prevention, it's important to note that unauthorized disclosure (even negligent, such as leaving documents unsecured) can result in administrative penalties (loss of clearance/job) or even criminal charges. This underscores why strict procedures exist and why personnel are trained repeatedly on how to handle sensitive information.

Understanding these classification concepts is foundational. It means that to secure sensitive work, one must not only deploy technical controls but also rigorously follow handling rules appropriate to the classification. The subsequent sections (on technical architectures and encryption) will build on this – for instance, using networks that are authorized for certain classification levels, or choosing encryption that meets the required level of protection (e.g., only NSA-approved algorithms for Secret and above). In summary: **know your information, its sensitivity, and the rules that apply to it** – this context informs every security decision that follows.

2

The Cybersecurity Policy Landscape

The U.S. Government operates within a robust framework of cybersecurity laws, directives, and standards. These policies set the expectations and requirements for how agencies must protect data and systems. A clear grasp of this landscape ensures that security measures align with what "the US government wants," from legislative mandates to executive initiatives.

- **Federal Information Security Modernization Act (FISMA):** Originally passed in 2002 and updated in 2014 (and again tweaked through the FY2022 NDAA), **FISMA** is a cornerstone law that requires each federal agency to develop, document, and implement an organization-wide program to secure information systems. Agencies must follow risk management practices and standards (largely defined by NIST) and report on compliance annually. FISMA essentially institutionalized the use of frameworks like the NIST Risk Management Framework for system authorization, and it holds agency heads accountable for

security. FISMA reports and scorecards are reviewed by Congress, making cybersecurity a matter of oversight. In practice, FISMA means that every federal system has to go through a rigorous security assessment and authorization and be continuously monitored.

- **Office of Management and Budget (OMB) Policies:** The OMB issues binding policy memoranda to agencies to implement laws and Presidential directives. For example, **OMB Memo M-22-09 (Jan 2022)** on *"Moving the U.S. Government Toward Zero Trust Cybersecurity Principles"* gave agencies a detailed roadmap and required outcomes for adopting zero trust by FY 2024. OMB also updates Circular A-130 (governing federal IT management) which includes cybersecurity requirements aligned with FISMA. Another notable memo, **OMB M-21-31 (Aug 2021)**, improved logging and incident response – it required agencies to log security-relevant events at defined levels and to protect those logs (a response to the SolarWinds breach where lack of logging hindered forensics). OMB memos ensure consistent implementation of cybersecurity across civilian agencies. They also often set deadlines (e.g., "enable MFA within X months") and require agency self-certifications or plans.
- **Cybersecurity and Infrastructure Security Agency (CISA) Directives:** CISA, under DHS, has authority to issue **Binding Operational Directives (BODs)** to civilian agencies for high-priority security actions. For instance, **BOD 22-01** (2021) established a requirement for agencies to remediate known exploited vulnerabilities within specific timeframes and created a public catalog of these vulnerabilities. This essentially forces agencies to patch

certain bugs (like those being actively used by attackers) in a timely manner. CISA also can issue Emergency Directives for urgent threats (like telling agencies to disconnect vulnerable servers during an active exploit campaign). Separately, CISA offers guidance and services – such as risk assessments and technical scanners – to agencies to help meet these policies.

- **National Cybersecurity Strategy (2023):** In March 2023, the White House released a comprehensive strategy outlining the vision and high-level goals for U.S. cybersecurity. It is built on **five pillars**: **(1)** Defend critical infrastructure, **(2)** Disrupt and dismantle threat actors, **(3)** Shape market forces to drive security and resilience, **(4)** Invest in a resilient future, and **(5)** Forge international partnerships. This strategy recognizes that beyond federal networks, we must raise security across critical sectors (energy, healthcare, etc.), tackle criminals (ransomware gangs), and shift the burden of security more towards software producers (e.g., holding vendors accountable for insecure software, which ties to pillar 3). While largely aspirational, it drives various implementing actions. For example, Pillar 4 (resilient future) explicitly includes advancing quantum-resistant encryption and digital workforce development – which aligns with topics covered in this book.
- **National Security and Defense Directives:** Apart from civilian agency guidance, there are policies for national security systems (Defense and Intelligence). **National Security Memoranda (NSMs)** and **DoD Directives** shape these. We saw NSM-10 (2022) for quantum (discussed later) and there's also **NSM-8 (2022)** on improving cybersecurity for National Security Systems which mirrored some of

EO 14028's requirements for the defense/intel world. The **DoD Zero Trust Strategy (2022)** and **DoD Instruction 8530** series for cyber defense are examples. Contractors working with DoD also face requirements via the Defense Federal Acquisition Regulations (DFARS) – e.g., DFARS 252.204-7012 mandates protection of Controlled Unclassified Info per NIST SP 800-171 (with self-attestation), and DFARS 252.204-7020/7021 roll out CMMC requirements.

- **Executive Order 14028 (2021) – "Improving the Nation's Cybersecurity":** This landmark EO came after major incidents (SolarWinds, Colonial Pipeline) and has driven many current initiatives. It required federal agencies (and their software suppliers) to enhance security in multiple ways: adopt **multi-factor authentication and encryption for data at rest/in transit** everywhere, implement **Endpoint Detection and Response (EDR)** solutions, accelerate movement to secure cloud services, and develop standards for **secure software development and supply chain**. It also established the mandate for a federal Zero Trust strategy (which became OMB M-22-09) and initiated things like requiring a Software Bill of Materials (SBOM) for software used by the government. EO 14028 has had perhaps the biggest impact on federal IT security since FISMA – a flurry of NIST guidelines, OMB policies, and agency actions have followed to meet its requirements.
- **Privacy and Data Specific Regulations:** While our focus is on classified/sensitive national security info, note that other rules protect sensitive data categories: **PII** is protected under the Privacy Act; **health data** under HIPAA; **credit card data** via PCI-DSS (industry standard agencies

follow for payments); etc. Agencies must juggle these too. There's increasing interplay between cybersecurity and privacy (hence NIST issued a Privacy Framework analogous to the Cybersecurity Framework). For classified info, privacy is less a concern, but for sensitive-but-unclassified data about citizens, privacy regulations strongly influence security controls (e.g., requiring data minimization and special handling for personal data).

- **Enforcement and Accountability:** The policy framework also establishes oversight mechanisms. Inspectors General review agency compliance with FISMA and OMB directives. Congress asks agency heads to testify on cyber posture. The administration, via OMB, can hold budgets or issue management admonishments if agencies lag (e.g., using the President's Management Agenda or cross-agency priority goals). On the contractor side, non-compliance with cyber requirements (like lying about 800-171 compliance) can lead to False Claims Act liability. So, these rules have teeth – they're not just suggestions, they're backed by audits, legal liability, and potential funding impacts.

In summary, the U.S. government's cybersecurity policies form a **hierarchy**: laws set broad requirements, executive orders and strategies set priorities, OMB and agency policies translate them into specific mandates, and NIST/CISA provide the standards and operational guidance to implement them. A government security engineer must ensure their programs comply with this mosaic – from high-level strategy down to technical configuration mandates – thereby achieving both security and compliance. Throughout this book, as we discuss

frameworks and practices, we will tie them back to these governing policies, ensuring the advice not only makes systems secure but also **meets the major current rules, plans, and regulations** that agencies are obligated to follow.

3

Frameworks for Security and Compliance

The mandates from laws and policies are operationalized through various **frameworks** – structured sets of standards, guidelines, and practices. These frameworks cover risk management, control implementation, and maturity measurement. They are the playbook by which agencies achieve security objectives in a consistent, auditable way. Below, we cover *"literally all the frameworks"* (or at least the most prominent ones) that the U.S. government employs for cybersecurity:

- **NIST Risk Management Framework (RMF):** The RMF is **mandatory** for federal information systems (under FISMA). It provides a **comprehensive, flexible 7-step process** for managing risk and achieving a security authorization to operate. The steps are: **Prepare** (organizational-level prep work), **Categorize** (the system's impact level for confidentiality, integrity, availability), **Select** (appropriate security controls from NIST SP 800-53 based on the

categorization and overlays), **Implement** (the controls), **Assess** (test the controls for effectiveness – often done by an independent assessor), **Authorize** (a senior official formally accepts the risk and grants an Authority to Operate, ATO), and **Monitor** (continuously oversee controls, perform periodic re-assessments). The RMF ties into a *"controls catalog"* (SP 800-53) and emphasizes risk-based decision-making. It's the backbone framework ensuring every system meets a baseline of security and that residual risks are known and accepted at the right level.

- **NIST Special Publication 800-53 (Security and Privacy Controls):** This extensive catalog contains the actual **security controls** that systems must implement. Controls are organized into families like Access Control, Incident Response, Media Protection, etc., and each control has a baseline for Low, Moderate, High impact systems. For example, a moderate system might require multi-factor authentication for network access (control IA-2) and a high system might require multi-factor for all privileged and non-privileged accounts plus more frequent credential changes. NIST periodically updates 800-53 to address new threats – e.g., recent Rev. 5 added controls for supply chain risk and IoT. As of 2025, Rev. 5 is in effect (Rev. 5.2 just released with minor tweaks). Federal systems select controls based on their category (FIPS 199 impact levels) and overlay any additional ones for specific needs (e.g., an overlay for intelligence systems, or one for privacy). **FedRAMP**, which is the cloud authorization program, also uses 800-53 controls (with an established baseline for cloud at different impact levels). So, 800-53 is truly *the* master list of safeguards – ranging from technical (encryption, audit

logs) to procedural (training, contingency planning).

- **NIST Cybersecurity Framework (CSF):** Published in 2014 (Rev. 2.0 expected soon), the **CSF** is a high-level, voluntary framework that many agencies and private companies use to assess and improve their cybersecurity programs. It organizes security activities into five **Functions – Identify, Protect, Detect, Respond, Recover** – which are further broken down into Categories and Subcategories of outcomes. For example, under Protect, one category is Access Control, with subcategories like "use MFA" or "least privilege". The CSF is not a replacement for RMF, but rather a complementary **management tool**. Agencies have mapped CSF to 800-53 controls to ensure they cover all basis. With CSF 2.0, there will be more alignment to governance and supply chain aspects. While not mandated by law, the CSF's language is often used in strategy documents and to communicate with non-technical executives about security posture ("How are we doing on Detect/Respond?" etc.). For a government contractor, aligning with CSF can also help meet requirements like CMMC or 800-171 by ensuring no major domain is overlooked.
- **NIST Special Publication 800-171 (Protecting CUI):** This framework applies to **non-federal systems** (e.g., contractors' networks) that handle **Controlled Unclassified Information (CUI)**. It has 110 security requirements mirroring a subset of 800-53 controls across 14 families – such as Access Control, Awareness and Training, Incident Response, etc. Examples include enforcing least privilege, incident reporting within 72 hours to the government, multi-factor auth for admins, encryption for CUI data in

transit, etc. Federal agencies include clauses in contracts requiring implementation of 800-171 if the contractor will handle CUI (like design drawings, or sensitive data provided by the government). Originally, compliance was self-attested, but due to patchy adoption, the DoD's CMMC program (Cybersecurity Maturity Model Certification) was created to *verify* and *certify* contractor compliance. In essence, **800-171 is the bridge** by which government security expectations propagate to industry partners.

- **CMMC (Cybersecurity Maturity Model Certification):** CMMC is specifically a DoD program, but it reflects broader federal priorities for supply chain security. **CMMC 2.0** (the streamlined version as of 2023) has three levels. *Level 1* (Foundational) is just 17 basic practices (mostly corresponding to FAR 52.204-21 basic safeguarding for Federal Contract Information). *Level 2* (Advanced) aligns exactly with NIST 800-171's 110 controls – this is for contractors handling CUIw. *Level 3* (Expert) will require some additional practices akin to 800-172 (which has more advanced, adversary-resistant controls). The key difference is **certification**: CMMC Level 2 will require a third-party assessment for most contractors, not just self-attestation. Starting late 2025, DoD will include CMMC requirements in solicitations, and by 2028 most CUI-handling contracts should require a Level 2 certification. This framework ensures that if you're doing sensitive work for DoD (like building a weapons system component), your own corporate IT is secured to government standards – no weak links for adversaries to exploit.
- **Identity, Credential, and Access Management (ICAM) Framework:** Managing user identities and their access

is so critical that separate guidance exists. The Federal ICAM architecture ensures agencies implement **HSPD-12** (homeland security directive 12) compliant PIV cards for physical and logical access, use federation standards for cross-agency access, and maintain robust identity lifecycle procedures. NIST SP 800-63 (Digital Identity Guidelines) is a framework for identity proofing and authentication assurance levels. Modern ICAM efforts dovetail with zero trust (e.g., **phishing-resistant MFA** using PIV or FIDO2 is now required many places). While not a single framework document, ICAM principles are embedded in OMB policies (like requiring use of PIV for all privileged access, etc.). In context of securing sensitive work, robust ICAM prevents unauthorized or unaudited access to systems – an essential layer of defense.

- **Software Supply Chain and DevSecOps Frameworks:** With EO 14028's focus on software security, frameworks like NIST's **Secure Software Development Framework (SSDF)** have become important. The SSDF (SP 800-218) outlines practices for secure coding, vetting third-party components, code review, testing, and integrity verification in the software development lifecycle. OMB memos (M-22-18 and its update M-23-16) now require agencies to obtain **attestations** from software vendors that they follow secure development practices per NIST. There's also guidance on **Supply Chain Risk Management (SCRM)** (NIST 800-161 as mentioned). While these might not be "frameworks" in the sense of RMF, they are structured sets of best practices agencies must enforce internally and via contracts.
- **Continuous Diagnostics and Mitigation (CDM) Pro-**

gram: This DHS/CISA program is a framework for continuously monitoring and managing cyber risk in civilian agencies. It provides tools and a dashboard for asset management, vulnerability management, identity management, and incident response. Agencies implement CDM capabilities to know **what's on their network, who's on their network, and what vulnerabilities exist** in near real-time. It's essentially a framework for operationalizing continuous monitoring (RMF Step 7) with metrics that can be rolled up to CISA and OMB. For example, CDM might track percentage of devices with up-to-date patches or how many endpoints have EDR installed, etc., and this feeds into FISMA metrics.

- **Other Sector-Specific Frameworks:** Depending on the agency's domain, additional frameworks may apply. The electric sector has **NERC CIP** standards (even federal power administrations must comply). Federal healthcare systems adhere to **HIPAA Security Rule**. Agencies managing credit data have to follow **FISMA High** controls plus **PCI-DSS** for cardholder data. While not all directly flow from "US government wants" as a whole, they are part of the constellation of requirements for protecting specific types of sensitive data. We mention them to acknowledge that a complete security program often means mapping and overlapping multiple frameworks. Fortunately, many controls overlap (e.g., if you do NIST 800-53 high, you've likely covered most HIPAA Security Rule requirements, etc.).
- **Maturity Models:** Beyond compliance checklists, agencies are also adopting maturity models to gauge how well they implement these frameworks. CMMC itself is a maturity

model for contractors. CISA's Cybersecurity Framework Profile assessments or internal agency assessments use maturity scoring (from ad-hoc to optimized) for areas like identity management or incident response. The goal is continuous improvement: not just "do we have a policy" but "is it effective and ingrained".

In practice, these frameworks are interconnected. For example, to authorize a system via RMF, you implement controls from 800-53; those controls might require you to use certain ICAM practices, align with CNSS (Committee on National Security Systems) policies for classified systems, etc. A key takeaway is that **compliance and security go hand-in-hand** in government – by following frameworks, you fulfill requirements *and* improve actual security. As we progress, we'll reference these frameworks (like citing specific control requirements or zero trust pillars from them) to illustrate how to practically achieve the security outcomes we discuss, ensuring both thoroughness and compliance with "what the US government wants."

4

Zero Trust Architecture in Government

One of the most transformative shifts in government cybersecurity in recent years is the move to a **Zero Trust Architecture (ZTA)**. Traditional security models operated on the concept of a trusted internal network (inside the perimeter) and untrusted external network (outside, e.g., the internet). Zero Trust upends this paradigm in response to sophisticated threats that easily bypass perimeter defenses (through phishing, stolen credentials, insider threats, etc.). As described by DoD, *"The foundational tenet of the Zero Trust model is that no actor, system, network, or service operating outside or **within** the security perimeter is trusted. Instead, we must verify anything and everything attempting to establish access."*. In simpler terms: **trust no one, authenticate and authorize continuously.**

Key principles of Zero Trust include **continuous verification, least privilege, device trust, and assuming breach**. Let's break down what that means for a government enterprise:

- **"Never Trust, Always Verify":** In a zero trust mindset, you assume the network is hostile. Every request for access – whether it's a user logging in to an application or one microservice calling another – must be authenticated, authorized, and encrypted. **Implicit trust based on network location is eliminated.** For example, just because a user is on a NASA internal network, they won't get access to a HR system unless their identity, device, and behavior are verified each time against policies.
- **Least Privilege Access:** This long-standing principle is supercharged in Zero Trust. Users (and programs) should have the minimum permissions necessary, and even when they have access, it might be time-bound or narrowly scoped. Zero Trust often employs *Just-In-Time* access (granting a privilege only for a limited time when needed) and *Just-Enough-Access* (granular roles). This way, even if an account is compromised, the adversary has limited reach. For instance, an attacker who steals a regular user's credentials should not be able to move laterally to sensitive data because that user's account has no business with those systems.
- **Device Posture and Integrity:** In Zero Trust, it's not only *who* the user is, but *what* device they're coming from and its security state. Every device – government-furnished laptop, BYOD mobile, server, IoT sensor – is assessed for trust. Is the device managed by the agency? Is it running approved security agents? Is its software up to date? This concept is often termed "**Device Trust**". Access can be denied or limited for devices that fail posture checks (e.g., quarantining a laptop that doesn't have the latest patches or has anomalous configurations). The

Continuous Diagnostics and Mitigation (CDM) device data feed into these decisions.

- **Micro-Segmentation:** Zero Trust encourages very fine-grained network segmentation, down to the level of isolating individual applications or workloads. Rather than a flat network where once inside you can reach many systems, each resource is protected. Architecturally, this might involve an internal gateway or software-defined perimeter for each app or service collection. For example, a database server would accept connections only from the application server that needs it, and even that connection would be authenticated and encrypted – not open to the whole internal subnet. This containment drastically reduces lateral movement possibilities.
- **Continuous Authentication & Authorization:** Unlike the old paradigm of "login once to the network or VPN, then you're in," Zero Trust requires continuous credential verification and policy evaluation. Modern implementations use short-lived tokens and frequent re-auth (behind the scenes). Technologies like **OAuth/OpenID Connect**, **SAML**, or **Kerberos** tokens might be used such that each service access requires a valid, unexpired token from an identity provider. Context is evaluated each time: *Has the user's role changed? Are they coming from a new IP? Is the device now showing signs of malware?* If something is off, the system can challenge for re-authentication or cut off the session. As OMB's Zero Trust strategy highlighted, agencies must *"continually verify each user, device, application, and transaction"* rather than a one-time check at the perimeter.
- **Visibility and Analytics:** Since Zero Trust generates

a lot of data (auth logs, network traffic, device posture updates), strong analytics are needed to make sense of it and detect anomalies. This involves aggregating logs in a SIEM (Security Information and Event Management) system and using automation. For instance, user behavior analytics might detect that an account which typically accesses one or two file shares is now trying to enumerate an entire server – flagging potential malicious activity. Anomalies can be addressed by automated responses (like forcing MFA challenge or device quarantine) in addition to alerting human analysts.

Now, how is the government implementing Zero Trust?

5

Implementing Zero Trust: Federal Initiatives and Components

The Federal Zero Trust Strategy (OMB M-22-09) set specific goals agencies had to achieve by end of FY 2024 across several pillars: Identity, Devices, Networks, Applications, and Data – aligning well with the image above and industry interpretations. Key required actions included:

- **Identity:** Enable **phishing-resistant multi-factor authentication** for all users (PIV or WebAuthn tokens) – basic OTP apps or SMS were deemed not sufficient. Consolidate identity systems so there's a single authoritative identity per user (many agencies had siloed directories). Implement enterprise-wide staff access systems (so users have one account to access all they need, reducing password reuse and shadow IT). User identities should be integrated with centralized analytics to detect unusual logins. Another mandate: use strong authentication for APIs (service accounts shouldn't use static passwords either – they should use certificates or token-based auth).

- **Devices:** Maintain a complete inventory of every device (including routers, VMs, IoT, etc.) and track their compliance (patched, configured securely). Use **endpoint detection and response (EDR)** tools on all endpoints to monitor for threats. Incorporate device health into access decisions – e.g., via a **Network Access Control (NAC)** system or cloud identity provider signals: if a laptop is missing a security agent or has out-of-date software, restrict its access. Encrypt data at rest on devices (using full disk encryption) and enforce remote wipe for mobile in case of loss.
- **Networks:** Implement **encrypted DNS (DNS over HTTPS) and encrypted HTTP (HTTPS with HSTS)** to prevent eavesdropping or manipulation of internal traffic. Segment networks and apply **TLS or IPsec encryption internally** – for example, require TLS 1.2+ for all web services, even those that traditionally might have been plaintext internally. Deploy **micro-segmentation** with tools like software-defined networks or cloud security groups so that systems only talk to what they need. Introduce **network intrusion detection** sensors to inspect traffic (though much will be encrypted, they can still analyze patterns or decrypted traffic at enforcement points). Plan for eliminating implicit trust of VPNs – agencies were told to start phasing out the idea that being on the VPN is enough. Instead, even VPN connections should be treated as untrusted until the user/device prove otherwise (and ideally, move to application-level authentication instead of network-level access).
- **Applications and Workloads:** Agencies need to **inventory all applications** including shadow IT, and secure

them by making them **accessible through encrypted, authenticated means** (often via the internet but gated by identity). One model is to put applications behind an **Application Proxy or Portal** – users authenticate to a zero trust access proxy which then connects them to the app if allowed, rather than users hitting the app directly on the internal network. Applications should undergo security testing (dynamic scans, penetration tests) and employ **modern protocols** (no old LDAP simple binds or outdated cryptography). The goal is that even legacy apps might be fronted by a modern identity-aware proxy to enforce zero trust. **Cloud adoption** helps here: many agencies are moving apps to cloud environments where identity-based access and micro-segmentation are easier to enforce uniformly.

- **Data:** Implement **automated data categorization and tagging**, and policies to detect and block unauthorized access or sharing of sensitive data. This includes deploying **Data Loss Prevention (DLP)** tools that monitor egress channels (email, uploads, USB copying) for sensitive content and stop it if policy-violating. Also, embrace concepts like **attribute-based access control (ABAC)** where access to data is controlled by fine attributes (e.g., classify documents and allow access only if a user's role, clearance level, and need-to-know attributes match the document's attributes). The Zero Trust strategy even envisioned agencies applying **user and entity behavior analytics (UEBA)** to data access – so if a user who usually reads 5 documents a day suddenly tries to download 1000 documents, it could be automatically halted as a likely stolen credential or malicious insider action.

- **Governance:** Agencies were also asked to create a Zero Trust implementation plan and assign a Zero Trust strategy lead. This ensures multi-year execution and integration with IT modernization efforts. The CIO Council and CISA provided reference architectures and maturity models (CISA's Zero Trust Maturity Model v2) to guide progress.

It's worth noting that **Zero Trust is not achieved overnight**. The year-2024 goals are a milestone, but full adoption is iterative. DoD, for instance, aims for a "**target**" state by 2027 where a robust Zero Trust implementation is in place department-wide (they defined 91 capabilities in their strategy to get there). Civilian agencies similarly will continue refining beyond initial compliance.

6

Challenges and Considerations:

Implementing Zero Trust in government faces challenges: legacy systems that can't easily support modern auth, budget constraints, workforce learning curve, and interoperability concerns (e.g., how to extend zero trust to inter-agency systems or to coalition partners securely?). There's also the user experience aspect – done poorly, security friction can impair productivity. Agencies have to balance strictness with practicality (e.g., maybe not every internal service needs re-auth for every access if risk is low, but high-value systems do – hence risk-based policies).

Another consideration is **encryption and inspection**. With "encrypt everything," agencies need new ways to inspect traffic for threat detection (like endpoint inspection, or decrypt at secure gateways for inspection then re-encrypt, etc.). It's a technical hurdle but solvable with proper design and tools.

Finally, **culture change** is a part of Zero Trust. Users and administrators must shift mindset from open connectivity to constrained, from owning "their network segment" to accepting enterprise guardrails. Training and communication, plus

strong leadership backing, are key so that people understand *why* these changes (like more logins or device rules) are necessary to protect the mission.

In conclusion, Zero Trust Architecture is a **foundational strategy for securely connecting users to the resources they need in a world of ever-present threats**. By treating every network transaction with skepticism and enforcing rigorous verification, agencies can significantly reduce the risk of unauthorized access or undetected breaches. Throughout this text, many other pieces (identity management, device security, encryption, monitoring) tie back into the Zero Trust approach – it truly is an **integrative framework** that brings together various security domains into a cohesive, modern defense. The U.S. government's strong push towards Zero Trust underscores its importance in achieving the highest levels of security for classified and sensitive work.

7

Post-Quantum Cryptography and Future Encryption

Cryptography is the bedrock of secure government communications – it protects classified data, secures military communications, and assures the integrity of information. However, a major paradigm shift is underway: the anticipated advent of **quantum computing** threatens to break most of the public-key cryptography in use today. This includes algorithms like RSA, Diffie-Hellman, and elliptic-curve cryptography (ECC) which are used for securing everything from confidential emails to HTTPS connections. A sufficiently powerful quantum computer could solve the mathematical problems (factorization, discrete logs) that these algorithms rely on, **rendering current encryption obsolete**.

To counter this, the field of **post-quantum cryptography (PQC)** – sometimes called quantum-resistant or quantum-safe cryptography – has been developing new algorithms believed to be secure against quantum attacks. The U.S. government, recognizing the strategic risk, is aggressively driving the adoption of PQC to protect sensitive information

well **before** quantum computers capable of cryptanalysis come online.

The Quantum Threat

- **Quantum Computers vs. Classical Encryption:** Traditional encryption's security is based on certain problems being computationally infeasible for classical computers. For example, RSA's security comes from the difficulty of factoring a large number (hundreds of digits long) into primes. A classical computer attempting brute force would take astronomical time. **Quantum computers**, using quantum bits (qubits) and algorithms like Shor's algorithm, can theoretically factor large numbers or compute discrete logarithms **exponentially faster** than classical computers. A rough estimate suggests that a cryptographically relevant quantum computer (one with thousands or millions of stable qubits) could break commonly used key sizes (e.g., 2048-bit RSA) perhaps in hours or days. Symmetric cryptography (like AES) is less affected – Grover's algorithm can speed up brute force searches, but doubling key lengths (e.g., AES-256) mitigates that. The real danger is to **public-key algorithms** that underpin key exchange and digital signatures.
- **Timeline of the Threat:** Experts differ on when a quantum computer capable of breaking RSA/ECC will be built – some say not for 10-15 years, others worry it could be sooner with unforeseen breakthroughs. The U.S. National Academies in 2022 suggested a 50% chance of such a quantum capability by 2037. It's speculative, but the prudent approach (and the government's stance) is to **act now**, given

the time it takes to transition crypto. **NSM-10** explicitly warned of quantum threats and initiated a coordinated government effort to migrate to PQC, highlighting that foreign adversaries are also pursuing quantum computing and we must not be caught unprepared.

- "**Harvest Now, Decrypt Later":** A critical point is that even if quantum decryption is a decade away, an adversary can **record encrypted data now** (say, intercepting sensitive encrypted traffic or stealing encrypted files) and store it until they have a quantum capability to decrypt it. This is particularly concerning for things like long-term classified secrets or personal data that has a long sensitivity lifetime. Thus, a classified communication intercepted today might be secret for 20 years – if a quantum computer is available in 10, that's within the window. This reality is a driver for agencies to adopt **quantum-resistant encryption as soon as practical**. NIST and NSA have been urging "don't wait" – for example, NIST's PQC project lead said "**We encourage system administrators to start integrating [the new algorithms] into their systems immediately**".

8

NIST's Post-Quantum Standards

The National Institute of Standards and Technology (NIST) led a global effort to identify and standardize quantum-resistant cryptographic algorithms:

- **The NIST PQC Competition:** Started in 2016, it invited cryptographers worldwide to submit candidate algorithms that could resist quantum attacks. Over 80 submissions were received, and through multiple rounds of analysis and cryptanalysis by experts, NIST narrowed the field to a few finalists by 2022. The criteria included security (against both quantum and classical attacks), performance, key sizes, and practicality.
- **Selected Algorithms:** In July 2022, NIST announced it would standardize **four algorithms** – for general encryption/key-establishment, the lattice-based **CRYSTALS-Kyber** was chosen; for digital signatures, **CRYSTALS-Dilithium** (lattice-based) and two alternatives, **FALCON** (lattice-based) and **SPHINCS+** (hash-based), were selected. Kyber and Dilithium were the

primary recommended algorithms due to strong security and efficiency, while FALCON and SPHINCS+ serve as backups/alternatives for certain use cases (FALCON for when smaller signatures are needed, SPHINCS+ for a totally different math basis — hashes — in case lattices ever have issues).

- **Standardization and Finalization:** Over 2023-2024, NIST drafted standards (FIPS 203 for Key Encapsulation – Kyber, FIPS 204 for Dilithium signatures, FIPS 205 for SPHINCS+ signatures) and opened them to public comment. In August 2024, **NIST released the first three finalized post-quantum cryptography standards**, meaning these algorithms are now formally standardized and can be implemented in government systems. The Kyber-based standard is now known as **ML-KEM (Module-Lattice Key Encapsulation Mechanism)**, Dilithium-based is **ML-DSA (Module-Lattice Digital Signature Algorithm)**, and SPHINCS+ is **SLH-DSA (Stateless Hash-based DSA)**. The FALCON-based standard (to be called **FN-DSA**) is expected later, likely 2025. These algorithms are designed so that **even a quantum computer cannot easily solve the underlying math problems**, which include lattice problems (like Learning With Errors) for Kyber/Dilithium /Falcon and hash-based one-way functions for SPHINCS+.
- **Comparing to Classical Crypto:** The new algorithms have different performance profiles. **Kyber** has relatively small public keys (800-**1000 bytes**) and ciphertexts, making it efficient for key exchange; it's also quite fast – comparable to or faster than RSA in many cases. **Dilithium** signatures are larger (a few kilobytes) compared to ECDSA, but still practical for many uses, and verification is fast. **SPHINCS+**

signatures are very large (tens of kilobytes) and slower, but it's included as a fail-safe since it relies on well-understood hash functions. Implementers are encouraged to use Kyber and Dilithium as the defaults and keep others as backup options. Importantly, these algorithms are believed to resist not just known quantum algorithms, but also traditional attacks (e.g., no one has found a classical attack that breaks them in feasible time either). NIST has stated that "**the algorithms [they] standardized are based on different math problems that would stymie both conventional and quantum computers.**" This dual requirement is crucial – we need algorithms secure against *all* adversaries, present and future.

- **Encouraging Adoption:** NIST and NSA are actively encouraging early adoption. NSA's **Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)**, announced in 2022, basically picks out Kyber and Dilithium (with certain parameter sets) as the intended future standards for classified and national security systems. However, it notes that until NIST finalized the standards and they're available in FIPS-validated implementations, agencies should continue using current approved algorithms (Suite B / CNSA 1.0). Now that NIST's standards are finalized (as of 2024), we expect NSA to issue guidance on when and how to transition NSS (National Security Systems) to CNSA 2.0 algorithms. NSA's stance has been to "**plan, prepare and budget**" for transition now, but not to roll out pre-standard algorithms on mission systems until vetted. As of late 2025, industry products (VPNs, browsers, etc.) are already starting to include PQC options (often in "hybrid" modes that combine classical and PQC algorithms for caution).

9

Government Transition Plans (NSM-10 and Beyond)

- **National Security Memorandum 10 (NSM-10):** Titled *"Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems"* (May 2022), this memo laid out the roadmap for U.S. government transition to PQC. It did a few key things: (1) It established a **Migration to PQC Project** at the National Cybersecurity Center of Excellence (NCCoE) to work with industry on practical migration guidelines. (2) It set the target of **2035** as the year by which current cryptography should be largely replaced by quantum-resistant cryptography in National Security Systems. (3) It emphasized **cryptographic agility** – agencies should build or update systems to be able to switch crypto algorithms more easily. (4) It mandated inventory and reporting: agencies had to report the status of their cryptographic inventory (where and what crypto is used) and their plans for migration. (5) It stressed interagency cooperation and

public-private partnerships, as the private sector will be delivering a lot of the tech needed for PQC.

- **OMB Memo M-23-02 (Migrating to Post-Quantum Cryptography):** This memo (Nov 2022) provides direction to civilian agencies in line with NSM-10. It set deadlines: by May 2023, agencies should inventory cryptographic assets; by Aug 2023, identify those that need PQC; by 2024-2025, begin testing implementations; etc. It also requires agencies to **prioritize high-value assets and high-impact systems** for early transition. For example, systems that handle sensitive data with long confidentiality requirements (nuclear secrets, citizen data) should move to PQC sooner. The memo also instructs agencies to train personnel on PQC and assess vendor readiness (many will rely on vendors to supply PQC-enabled products).
- **CNSA Suite 2.0 Timeline:** The NSA indicated that once NIST standards are ready (which they now are as of 2024), they will require their use in NSS after a transition period. NSA's FAQ suggests a multi-year transition where by around 2030, new systems should be using PQC and by 2035, legacy crypto should be phased out. Notably, **classified information with truly long lifetimes** (that must stay secret for decades) might even need additional protections like quantum key distribution (QKD) or one-time pads, but those are specialized cases.
- **Implementing in Practice:** For federal agencies, migrating to PQC means updating or replacing many cryptographic components: VPN software, internal PKI (public key infrastructure) systems, secure email, identity smartcards (PIV cards use RSA/ECC today – NIST is working on updating FIPS 201 standards for PIV to support PQC

algorithms), etc. It's a labor-intensive process. Agencies will likely start by enabling **hybrid modes** (e.g., TLS can use both an RSA and a PQC key exchange simultaneously – so even if RSA is broken, the session is still protected by the PQC key). This provides confidence during an interim period. Eventually, pure PQC modes will be used.

- **Challenges:** PQC algorithms have **larger keys and signatures**, which might impact network bandwidth and storage. They may also be slower in certain operations or require more memory. Devices with constrained resources (smartcards, IoT sensors) might struggle to implement them without hardware upgrades. Interoperability is another challenge – a secure email system, for instance, only works if the sender and receiver both support the new algorithms. Thus, transition will be a coordinated dance; interim compatibility (maybe keeping both old and new crypto available) is necessary. Agencies are testing these things now (NCCoE has demonstration projects for, say, a PQC-enabled VPN or a PQC enterprise TLS).
- **Ensuring Security of PQC:** It's worth mentioning that while PQC eliminates known quantum attacks, the new algorithms must still withstand classical attacks and implementation bugs. As these get deployed, we may discover new vulnerabilities (e.g., side-channel attacks on a PQC algorithm's implementation). Thus, defense-in-depth remains crucial: just because you have PQC doesn't mean you stop monitoring or patching. NIST will also continue a "**Round 4**" process to standardize more algorithms as backups, and the community will keep cryptanalyzing the current ones. There's a chance (hopefully small) that an approved algorithm might be broken or weakened in the

future, so cryptographic agility – the ability to swap it out quickly – is very important (hence all the emphasis on agility frameworks).

In essence, **post-quantum cryptography is about future-proofing the confidentiality and integrity of sensitive information**. The U.S. government's proactive stance (standardizing early, mandating migration plans) aims to avoid a scenario where an adversary's quantum leap suddenly exposes years of secrets. Just as the government led in adopting encryption decades ago (like going from DES to AES, or adopting public-key infrastructures), it is now leading in this quantum-resistant overhaul. Agencies that deal with classified or otherwise sensitive data should closely follow NIST/NSA guidance and begin weaving PQC into their upgrade cycles. By the time a powerful quantum computer arrives – whether in a decade or tomorrow – our goal is that all critical data is already protected by algorithms that such a computer **cannot break**, thus preserving the security and confidentiality that are fundamental to government operations.

10

Secure Systems Engineering and Architecture

Thus far, we've discussed high-level frameworks and emerging technologies. We now turn to the **practical engineering** of secure systems – how to design and configure the IT infrastructure (hardware, software, networks) in a government environment to be robust against threats. This includes on-premises systems, cloud services, networking, hardware devices, and the interconnections between them. In an age where remote work and cloud are increasingly common, "engineering everything locally securely" means applying security controls consistently across all environments and ensuring that wherever data or processes reside (local datacenter, field office, or cloud), they are protected to the required level.

Secure Configuration Baselines

One of the fundamental steps in system security is establishing **secure configuration baselines** for all systems:

- **Hardening Systems:** Government systems should be configured with security in mind from the start. This involves disabling unnecessary services and ports, removing default accounts or passwords, applying the principle of least functionality. The DoD's **Security Technical Implementation Guides (STIGs)** and the Center for Internet Security (**CIS Benchmarks**) are key resources. For example, a Windows Server STIG will prescribe dozens of settings: enable Windows Firewall, enforce password complexity, audit certain events, disable old protocols like SMBv1, etc. When setting up a new server or image, using these guides ensures a hardened starting point.
- **Automated Configuration Management:** Agencies typically use automated tools (like Microsoft Group Policy, Ansible, or configuration management systems) to enforce baseline settings and detect drift. SCAP (Security Content Automation Protocol) benchmarks exist for many STIGs, allowing automated scanning to see if a system deviates. Continuous compliance monitoring is part of continuous ATO processes – ensuring that systems remain in their approved, secure state and flagging if, say, someone enabled an insecure option.
- **Patch Management:** A secure baseline is not static; **patching** is critical to fix newly discovered vulnerabilities. Agencies maintain patch management programs to test and deploy OS, application, and firmware updates regularly

(often monthly for routine patches, faster for critical zero-days). Given the government's scale, this can be challenging – but directives like CISA's BOD 22-01 (with its Known Exploited Vulnerabilities catalog) have lit a fire under agencies to patch expeditiously. Many agencies use centralized systems (WSUS, SCCM, or cloud services) to push patches and then verify compliance.

- **Least Privilege and Access Controls:** Engineering a secure system includes setting up accounts, privileges, and file permissions such that each user or process only can access what's necessary. This extends from Active Directory group design (ensuring admin privileges are segmented and monitored) down to file system ACLs and database roles. Default admin accounts are renamed or disabled if possible, and day-to-day operations are done with non-privileged accounts. Privileged Access Management (PAM) systems may be employed to control use of admin accounts (checking them out, rotating passwords, etc.).

By having strong baselines, even if an attacker gets in, they find fewer weaknesses to exploit for escalation or persistence (e.g., if PowerShell logging is enabled per baseline, their activities are recorded; if no guest account exists, they can't hide behind it).

11

Network Architecture and Segmentation

When we talk about "engineering everything locally securely", network design plays a huge role – especially in government facilities (like data centers, offices, and tactical environments):

- **Layered Networks (Zoning):** A common practice is to define network zones by sensitivity. For instance, an **Internal network** (for general user workstations), a **DMZ** (for public-facing servers accessible from the internet), a **Restricted network** (for servers/databases with sensitive data), etc. Firewalls or internal routers strictly control traffic between zones – only allowing necessary flows (based on source, destination, port, sometimes user). So, a user PC can reach the email server, but not directly query the database server, etc. For classified environments, the entire network is effectively a "restricted" zone, but even within it, further segmentation occurs (as discussed in Zero Trust section).

- **Air-Gapped Systems:** Truly critical systems might be **physically isolated** with no network connection to less secure networks or the internet. For example, vote tallying systems, or intelligence analysis networks, or nuclear command and control networks, are either entirely offline or connected only through one-way diodes or tightly guarded gateways. Many classified networks are *air-gapped from the internet* – e.g., SIPRNet has no direct connection to NIPRNet (unclassified internet-facing network). If data must cross, it goes through a *Cross Domain Solution (CDS)*, which is a specialized guard that filters and sanitizes data (for instance, to email a Secret document to an Unclassified email, a human-reviewed sanitization might be required via the guard).
- **Secure Communication Channels:** Within a facility, sensitive data traveling between systems should use secure protocols. This could mean **internal TLS** for web services, SSH for remote admin, or IPsec at the network layer. Government encryption standards (like using FIPS 140-validated cryptographic modules and FIPS-approved algorithms) ensure that even inside the firewall, data transmissions are protected from interception. **Network encryption** at layer 2 or 3 is also used – for example, NSA Type 1 encryptors might sit at network junctures to encrypt all traffic leaving a secure area, even if it's going over a dedicated fiber to another building.
- **Intrusion Detection/Prevention Systems (IDS/IPS):** Strategically placed sensors in the network (at internet gateways, between internal segments) monitor traffic for malicious signatures or anomalies. In a secure architecture, if an attacker does manage some network penetration, these

systems serve as tripwires. Modern implementations might include **Network Traffic Analysis (NTA)** using AI to catch things like beaconing or unusual lateral movement. All network logs feed into a SIEM for correlation with host logs.

- **Software-Defined Networks (SDN) and Cloud Virtual Networks:** As agencies move to cloud or modernize data centers, they increasingly use SDN capabilities to enforce network segmentation in a more dynamic way. For example, in a cloud environment like AWS GovCloud, one can use separate VPCs for different apps or data types, and security groups as virtual firewalls controlling traffic at instance-level. These are engineered analogously to on-prem segmentation but allow more granularity and automation (infrastructure as code, etc.). The principle remains: **minimize open pathways** and make those that exist very explicit and controlled.
- **Resilience and Redundancy:** A secure system must also be reliable. Engineering considerations include redundant network paths, failover for firewalls, backup power supplies – because a critical system going down can also be a security issue (impacting availability). Designing for high availability and quick recovery (e.g., having hot standby sites, regularly tested backups) is part of secure engineering, especially under frameworks that emphasize resilience (like the National Cyber Strategy's pillar on resilience).

12

Hardware Security and Peripheral Controls

Hardware forms the foundation, and we must ensure hardware itself and its interfaces are secure:

- **Trusted Platform Module (TPM) and Secure Boot:** Modern computers often include a TPM chip which can securely store cryptographic keys and measure system integrity (checking hashes of bootloaders, etc.). Enabling **Secure Boot** ensures the BIOS/UEFI only loads signed bootloaders and OS kernels. This prevents rootkits or bootkits from hiding. Government workstation and server baselines typically mandate Secure Boot ON and, if possible, using the TPM to enforce that the OS has not been tampered with. There's also the concept of remote attestation (a device proving its integrity to a network controller) that may grow in use for zero trust device posture validation.
- **Anti-Tamper Protections:** Particularly for deployed systems (like a router in an embassy or a

device in the field), one worries about physical tampering. Hardware can include anti-tamper sensors – case open detection, or epoxy coating on chips to prevent modification. If triggered, systems can wipe sensitive data or at least log the event. In less exotic form, even server racks have lockable doors and cameras watching them in data centers to deter anyone from opening them without clearance.

- **USB/Peripheral Control:** One of the common vectors in breaches is USB drives or peripherals (remember Stuxnet and the infected USB that introduced malware to an air-gapped network). Agencies implement policies and technical controls for this: disabling autorun features, whitelisting USB devices by serial (using software like Device Control), or completely blocking mass storage via group policy. Some high-security environments epoxy-fill or physically block USB ports. If removable media use is needed (e.g., to transfer data into an isolated network), there are strict procedures: scanning the media on an offline scanner PC, using only agency-issued encrypted USB drives, etc. Additionally, "**Rubber Ducky**" attacks (malicious USB pretending to be keyboard) are mitigated by disabling new HID devices or alerting when a new type of peripheral is connected.
- **Hardware Lifecycle:** Engineering security includes the whole lifecycle: procurement (choosing trusted suppliers as discussed in supply chain

section), secure installation (sanitizing and configuring hardware out of the box – e.g., updating firmware to latest non-vulnerable version), maintenance (monitoring for firmware updates or part recalls), and disposal (sanitizing or destroying storage components). For classified systems, disposal is critical: drives are degaussed and shredded; even devices like printers or multifunction devices which may store data are cleansed or destroyed.

- **Physical Placement and Emissions:** Certain hardware considerations, like where you place wireless access points or how you shield cables, matter to security. For example, on a classified network, one typically does not use standard Wi-Fi at all (because it's a radio emission). Any wireless needed must have NSA-approved encryption and often must be turned off in a SCIF. Even electromagnetic emanations (TEMPEST concerns) might require using shielded cables, filters, and distance controls for equipment processing classified information to mitigate eavesdropping risk.

13

How to Air-Gap Systems for TEMPEST, Signal Security, and the Weird Physics of Leaking Secrets

When people say "cybersecurity," they usually mean software. TEMPEST is the reminder that the universe does not care about your threat model. Electrons wiggle, wires radiate, components couple into each other, and your "secure" system can accidentally whisper its secrets into the room as electromagnetic (EM) energy, conducted noise on power lines, optical leakage, acoustic vibrations, or even subtle ground currents.

TEMPEST (often used as a catch-all term) refers to controlling and reducing **compromising emanations**, meaning unintended signals that can reveal information.

This chapter is defensive. It is about preventing leakage, testing your environment, and building systems that do not spray sensitive data into the surrounding physics.

1. What TEMPEST actually means

Any system that processes information moves energy around. That energy has side effects:

- **Radiated emissions**: unintended RF energy from cables, traces, monitor connectors, keyboard matrices, GPUs, memory buses.
- **Conducted emissions**: noise riding on power lines, ground, Ethernet shields, and shared infrastructure.
- **Cross-coupling**: one cable acts like an antenna, another cable acts like a receiver, and your rack becomes a gossip network.
- **Optical leakage**: LEDs, display backlight modulation, optical carriers in fiber with poor isolation (rare, but real).
- **Acoustic and vibro-mechanical leakage**: coil whine, fan harmonics, keystrokes, even resonances that correlate with workload.
- **Thermal side channels**: heat patterns over time, especially in tightly packed compute (more relevant for local attackers).

The brutal truth is:

If a physical phenomenon correlates with the data, it can become a side channel.

2. Threat model: who can exploit signal leakage?

You do not need to assume movie-villain attackers for TEMPEST controls to make sense. The threat tiers look like this:

Tier 0: accidental exposure

- Interference causes malfunctions, data corruption, dropped packets, sensor noise.
- Neighboring systems leak operational details (workload timing, keyboard activity, screen redraw patterns).

Tier 1: opportunistic observers

- A nearby device or environment picks up signals unintentionally (bad cable routing, shared power strips, poor grounding).
- "Corporate espionage by laziness" is common: someone places sensitive gear near windows, shared walls, or unfiltered wiring.

Tier 2: deliberate but practical adversary

- Someone in the building, parking lot, adjacent suite, or shared utility infrastructure focuses on emissions.
- Goal: recover content, infer activity, or fingerprint systems.

Tier 3: well-resourced adversary

- Specialized receivers, calibrated antennas, and patient analysis.
- These are the cases that drive strict standards and facility controls.

Your defenses should be proportional. But also remember: TEMPEST defenses often improve reliability and reduce weird bugs, so you get engineering value even in low-threat settings.

3. The physics in one page

Two ideas explain 90 percent of the problem:

3.1 Coupling paths

A signal leaks from source to observer through one or more paths:

- Capacitive coupling (electric field)
- Inductive coupling (magnetic field)
- Radiated far-field coupling (antenna-like behavior)
- Conducted coupling (through power and ground)
- Common-impedance coupling (shared return paths)
- Cable-to-cable coupling (crosstalk)

A practical mental model:
Source -> coupling mechanism -> path -> receiver
Fix any one link and you reduce risk.

3.2 Why "digital" still leaks like analog

A digital edge is fast. Fast edges contain high-frequency components.

A simple rule of thumb:

- Faster rise time means more high-frequency energy.
- More high-frequency energy means more radiation, more coupling, and more chance of leakage.

If you want a rough "frequency content" scale:

- bandwidth_hz is approximately 0.35 / rise_time_seconds

Example:

- rise_time = 1 ns
- bandwidth_hz about 0.35 / 1e-9 = 350 MHz

That is why a "100 MHz" bus can create energy far above 100 MHz.

4. What leaks in real systems

4.1 Displays and video links

Displays are classic leakage sources because they carry structured data. Modern digital display links are better than old analog VGA, but high-speed serial lines still radiate if shielding and grounding are sloppy.

Defensive posture

- Prefer fiber-based display extension or well-shielded, certified cables.
- Minimize cable length.
- Route away from edges of secure areas.
- Use physical zoning (later section).

4.2 Keyboards and input devices

Keyboards can leak via cable emissions, wireless protocols, and even subtle power draw changes. Wireless peripherals add attack surface and can make compliance harder.

Defensive posture

- Use wired peripherals in sensitive spaces.
- Apply port control and device allowlisting.
- Keep input cables short and separated.

4.3 CPUs, GPUs, memory, and switching regulators

Compute load changes power draw. Switching regulators (VRMs) are noisy by nature and can imprint workload patterns on power rails.

Defensive posture

- Use high-quality PSUs with filtering.
- Add line filters where appropriate.
- Physically separate "red" processing zones from general office zones.
- Pay attention to grounding and bonding.

4.4 Networking and common-mode emissions

Ethernet is differential, but common-mode currents on shields and poorly bonded grounds can radiate.

Defensive posture

- Use properly terminated shielded cabling only when bonding is correct.
- Use fiber for long or boundary-crossing runs.
- Avoid routing network and power tightly together for long distances.

5. Signal security is bigger than TEMPEST

TEMPEST is one slice of **signal security**. A complete signal-security view includes:

- Emissions security (TEMPEST style)
- Communications security (encryption, key management)
- Transmission security (anti-jam, anti-spoof, physical link hardening)
- Spectrum management (knowing what is in your RF environment)
- Supply chain integrity (components, cables, firmware)
- Physical security and zoning (who can place a receiver where)

A common failure mode is to do encryption perfectly and ignore that the workstation is radiating structured signals into the nearest wall.

6. Zoning and RED BLACK separation

The most practical control framework is **separation by zone**.

- **RED**: equipment and signals that carry sensitive plaintext.
- **BLACK**: equipment and signals that are encrypted or non-sensitive.

Core principles:

1. Do not run RED and BLACK cabling together.
2. If they must cross, cross at 90 degrees and minimize

parallel runs.
3. Maintain physical distance. Distance is the cheapest shield.
4. Use dedicated power circuits where possible.
5. Filter where signals leave a secure zone.

Think of it as preventing a "side-channel supply chain" from forming inside your building.

7. Shielding, bonding, grounding: the holy trinity (and how it goes wrong)

These words cause endless arguments because people mix them up.

- **Shielding**: blocking fields using conductive enclosures or cable shields.
- **Bonding**: making low-impedance connections between metal parts so they do not float and radiate.
- **Grounding**: providing a reference and safety path for fault currents, and managing return paths for signals.

Most real-world leakage disasters come from:

- Shields connected on one end when they should be bonded on both ends (or vice versa, depending on design).
- High-impedance "ground" paths causing common-mode currents to radiate.
- Daisy-chained power strips creating shared impedance coupling.
- Cabinet doors or panels with poor electrical continuity (paint, oxidation, loose fasteners).

Practical defensive rules:

- Keep bonds short, wide, and mechanically solid.
- Use star-grounding concepts for sensitive analog when appropriate, but do not cargo-cult it into high-frequency digital without thinking.
- Treat cable shields as part of an RF system, not as decoration.

8. Filtering and isolation controls

8.1 Power line filtering

If sensitive processing happens on a machine, the power draw can carry correlated noise.

Defensive options:

- EMI filters at equipment input
- Power conditioners designed for EMI suppression
- Isolation transformers (use carefully, understand safety and leakage currents)
- Dedicated circuits and clean grounding

8.2 Data line filtering

For boundary crossings:

- Fiber is your friend.
- Use approved filters for lines leaving secure zones.
- Prefer optical isolation for control signals in mixed environments.

8.3 Physical isolation

If you can place sensitive systems away from exterior walls, windows, shared corridors, and neighboring tenants, do it. This is boring, and boring is secure.

9. Operational controls: how organizations fail in practice

TEMPEST is often defeated by "ops reality":

- Someone replaces a short certified cable with a cheap long one.
- Someone adds a KVM extender to fix convenience.
- A contractor reroutes a power cable next to a data bundle.
- An "IT closet" becomes a storage room full of random metal objects and coiled cables.
- A new neighbor moves in next door with unknown equipment.

Defensive posture:

- Configuration management for cabling and peripherals.
- Inspections after changes.
- Clear labeling of RED vs BLACK.
- Procurement rules for cables, adapters, and peripherals.
- Training that explains "why," not just "no."

10. Testing and verification: you cannot secure what you do not measure

You do not need a spy-lab to benefit from measurement. Even basic RF hygiene testing can catch the obvious problems:

- Are emissions worse near a certain cable run?
- Does a monitor cable radiate more than expected?
- Does moving the system away from a wall reduce leakage potential?
- Are there unexpected narrowband emitters inside your secure area?

10.1 A defensive RF survey workflow (high-level)

1. Define the sensitive zone boundaries.
2. Identify likely emitters: displays, cables, compute racks, PSUs.
3. Sweep the environment for unexpected energy.
4. Compare "idle" vs "active workload" profiles.
5. Fix routing, shielding, filtering.
6. Re-test and document.

11. Example: a simple defensive spectrum survey script (Python)

This is a **defensive measurement scaffold**. It helps you capture spectrum snapshots and compare changes. It does not attempt to recover content. It just measures energy levels over frequency so you can spot suspicious peaks and reductions after mitigation.

It assumes an RTL-SDR style receiver and pyrtlsdr. If you have a different SDR, the structure is the same.

```
"""
Defensive RF survey tool (TEMPEST hygiene)
- Sweeps a frequency range in steps
- Captures a power estimate per bin
- Saves results to CSV for before/after comparison

Notes:
- This is for emissions awareness and remediation
verification.
- Do not use this to attempt content reconstruction.
"""

from dataclasses import dataclass
from typing import List, Tuple
import time
import csv
import math

import numpy as np

try:
    from rtlsdr import RtlSdr
except ImportError as e:
```

```
    raise SystemExit(
        "Missing dependency: pyrtlsdr. Install
        with:\n"
        "  pip install pyrtlsdr\n"
        "Also ensure librtlsdr is installed for your
        OS.\n"
    )

@dataclass
class SweepConfig:
    f_start_hz: float
    f_stop_hz: float
    step_hz: float
    sample_rate_hz: float = 2.4e6
    center_settle_s: float = 0.15
    num_samples: int = 256 * 1024
    gain_db: float = 20.0
    window: str = "hann"
    averages: int = 3

def window_fn(name: str, n: int) -> np.ndarray:
    name = name.lower().strip()
    if name == "hann" or name == "hanning":
        return np.hanning(n)
    if name == "blackman":
        return np.blackman(n)
    if name == "rect":
        return np.ones(n)
    raise ValueError(f"Unknown window: {name}")

def power_spectrum_db(samples: np.ndarray, win:
np.ndarray) -> Tuple[np.ndarray, np.ndarray]:
    x = samples.astype(np.complex64) * win
    X = np.fft.fftshift(np.fft.fft(x))
    p = (np.abs(X) ** 2) / len(X)
    p_db = 10.0 * np.log10(p + 1e-12)
    freqs = np.fft.fftshift(np.fft.fftfreq(len(X),
```

```
    d=1.0))
    return freqs, p_db

def sweep(cfg: SweepConfig) -> List[Tuple[float,
float]]:
    sdr = RtlSdr()
    sdr.sample_rate = cfg.sample_rate_hz
    sdr.gain = cfg.gain_db

    n = cfg.num_samples
    win = window_fn(cfg.window, n)

    results: List[Tuple[float, float]] = []

    try:
        f = cfg.f_start_hz
        while f <= cfg.f_stop_hz:
            sdr.center_freq = f
            time.sleep(cfg.center_settle_s)

            # Average a few captures for stability
            p_accum = None
            for _ in range(cfg.averages):
                samples = sdr.read_samples(n)
                rel_freqs, p_db =
                power_spectrum_db(samples, win)
                if p_accum is None:
                    p_accum = p_db
                else:
                    p_accum += p_db

            p_mean = p_accum / float(cfg.averages)

            # Summarize: take the max power in this
            capture window as a simple "peakiness"
            indicator
            peak_db = float(np.max(p_mean))
```

```
            results.append((float(f), peak_db))

            print(f"center_hz={f:.0f}
            peak_db={peak_db:.2f}")
            f += cfg.step_hz

    finally:
        sdr.close()

    return results

def save_csv(path: str, rows: List[Tuple[float,
float]], cfg: SweepConfig) -> None:
    with open(path, "w", newline="") as f:
        w = csv.writer(f)
        w.writerow(["center_hz", "peak_db",
        "sample_rate_hz", "num_samples", "gain_db",
        "averages", "timestamp_unix"])
        ts = int(time.time())
        for center_hz, peak_db in rows:
            w.writerow([center_hz, peak_db,
            cfg.sample_rate_hz, cfg.num_samples,
            cfg.gain_db, cfg.averages, ts])

if __name__ == "__main__":
    cfg = SweepConfig(
        f_start_hz=30e6,
        f_stop_hz=300e6,
        step_hz=2.0e6,
        sample_rate_hz=2.4e6,
        num_samples=256 * 1024,
        gain_db=20.0,
        averages=3,
    )

    rows = sweep(cfg)
    save_csv("rf_sweep_results.csv", rows, cfg)
```

```
    print("Saved: rf_sweep_results.csv")
```

How to use it defensively:

- Run a baseline sweep with the system off or idle.
- Run again with the system under typical workload.
- Apply mitigations (cable routing, shielding, filters).
- Re-run and compare peaks and overall behavior.

This is not a certification test. It is an engineering flashlight.

12. Design patterns that drastically reduce risk

If you want a compact list of "actually works" patterns:

- **Fiber across boundaries**: for network, for display extension, for anything leaving a controlled space.
- **Short cables, good cables**: minimize antenna length.
- **Distance and zoning**: put sensitive systems deeper inside controlled areas.
- **Filtered power entry**: treat power as a data path unless proven otherwise.
- **Enclosures that bond well**: doors, panels, seams matter.
- **Reduce edge rates where feasible**: series resistors, proper termination, controlled impedance.
- **No ad-hoc adapters**: cheap dongles are physics wildcards.
- **Document and re-verify after changes**: emissions security is configuration management plus physics.

13. Implications for modern "secure compute" and AI workloads

AI systems add special considerations:

- **High dynamic loads** (GPU boosting, VRM transients) can create strong conducted signatures.
- **Dense cabling** (multiple high-speed links) increases coupling opportunities.
- **Large power infrastructure** (PDUs, redundant PSUs) creates more paths for noise to propagate.

Defensive approach for AI racks:

- Isolate power feeds for sensitive racks.
- Prefer fiber networking, especially for boundary crossings.
- Separate management networks physically and logically.
- Use consistent grounding and bonding across the rack line.
- Treat every cable bundle as a potential antenna array.

14. Policy and compliance angle (practical, not mystical)

Many organizations map TEMPEST controls into:

- Facility security requirements
- Procurement rules (certified equipment where required)
- Zoning and separation rules (RED BLACK)
- Inspection and change control procedures
- Periodic assessments and measurement

The key idea is governance that respects physics:

- "No unapproved cable changes" is not bureaucracy. It is preventing silent reintroduction of leakage paths.

15. A pragmatic checklist

Physical layout

- Sensitive processing away from exterior walls and public spaces
- Controlled room access
- Clear zoning boundaries and labeling

Cabling

- Short, shielded where appropriate, properly bonded
- RED and BLACK separation maintained
- No random extenders, adapters, or undocumented KVMs

Power

- Quality PSUs, filtered entry where needed
- Avoid shared strips and long daisy chains
- Documented grounding and bonding

Devices

- Limit wireless peripherals
- Control LEDs and indicators where they could encode state
- Prefer hardened endpoints for sensitive workflows

Verification

- Baseline and post-change RF hygiene sweeps
- Documentation of mitigations and retest results

Closing thought

TEMPEST is what happens when your computer accidentally becomes a radio station for its own internal thoughts. The solution is not paranoia. It is engineering: control coupling, control paths, control boundaries, and measure the real world instead of assuming it behaves.

If you keep doing that, the universe gets quieter, your systems get more reliable, and your secrets stop trying to escape through the walls like bored ghosts made of electrons.

14

Secure Development and Local Applications

Often agencies develop custom applications or scripts locally. Secure engineering extends to the software development practice (some of which overlaps with supply chain security earlier):

- **Secure Coding:** Developers should follow secure coding standards (like OWASP Top 10 for web apps or SEI CERT coding standards for C/C++). This includes validating inputs (to prevent SQL injection, etc.), using parameterized queries, avoiding use of deprecated functions, proper error handling, and logging. Many agencies require static code analysis and security testing as part of the development pipeline (DevSecOps). A local development environment should itself be secure – using non-privileged accounts, scanning code repositories for secrets (to ensure API keys or passwords aren't accidentally hardcoded and pushed).
- **Segregation of Environments:** Keep development, testing, and production environments separate. If developing

locally, ensure that test data does not include real sensitive information unless the dev environment is secured equivalently to production. It's better to use sanitized or synthetic data for testing. Access to source code and build systems should be restricted to the dev team, and those systems should be monitored (since an attacker who can insert a backdoor in source code or a CI/CD pipeline could compromise the end product).

- **Secure APIs and Microservices:** Modern architectures use a lot of APIs. Ensure APIs have proper authentication (no relying solely on network location). Use tokens with appropriate scope. Implement rate limiting and input validation at the API gateway or service level to thwart misuse. Internal microservices might use mutual TLS so they authenticate each other. Basically, apply zero trust at the application inter-service level as well.
- **Local Administrative Interfaces:** Many systems have admin interfaces (like router web UIs, management consoles). These should be on dedicated management networks or accessible only via jump hosts – not on general user subnets or the internet. Default credentials must be changed, and ideally, centralized auth (like RADIUS/TACACS+ with AD credentials) used for infrastructure devices, with unique accounts per admin (no shared "admin" login). Role-Based Access Control (RBAC) should restrict what each admin can do to prevent accidents or misuse.

15

Monitoring and Incident Response Preparedness

Secure engineering also means building in monitoring hooks and designing with incident response in mind:

- **Comprehensive Logging:** Systems should be configured to log security-relevant events (and in a tamper-evident way). This includes OS logs (login attempts, file access if sensitive, process execution for whitelisting violations, etc.), application logs (especially auth events, data access), and network device logs (firewall accept/deny, VPN connects). Use centralized logging solutions (SIEM) to aggregate logs for correlation. Importantly, ensure logs of sensitive systems are themselves protected (restrict log server access, encrypt logs in transit/storage, and retain them per policy). OMB's guidance post-SolarWinds (M-21-31) required agencies to achieve **logging maturity level 3 or 4**, meaning robust, centralized logs retained for months and actively used.
- **Built-in Intrusion Detection:** Besides network

IDS, hosts can have HIDS (Host Intrusion Detection Systems) or EDR that detect unusual behavior. These agents often feed alerts to the SOC. Engineering choice: ensure these agents are deployed on all systems and tuned not to break functionality but also not to be too lax. Use canaries or honeypots in the network (like an unused server with a fake dataset; any access to it is likely a sign of intruder).

- **Incident Response Hooks:** In architecture, think "If this system was compromised, how would I detect and contain it?" That leads to design decisions like: having an out-of-band network for management (so you can still access devices if the main network is flooded or compromised), having jump boxes for admin access that can be closely monitored, segmenting critical backups on a network that can be isolated if ransomware hits the main network, etc. It also means regularly testing incident response (e.g., simulate a breach and see if monitors trigger and how well containment works).
- **Backups and Recovery:** Ensure regular **backups** of critical data and configurations are taken, stored securely (encrypted, and offline or in immutable storage to prevent attackers from deleting them). Test restoring backups periodically. In government, data integrity is as important as confidentiality – an attacker altering data is a big threat, so having authoritative backups and checksums to restore trusted state is key. For classified data, backups must be stored at equal classification and protected as such.

When systems are engineered with these considerations, the overall security posture is greatly improved. Many successful attacks exploit *misconfigurations* or *omissions* rather than fundamental flaws – e.g., a server left with default credentials, a network segment with an open port that wasn't filtered, a known patch not applied, logging turned off, etc. Secure engineering is about systematically eliminating those gaps. It's a detailed, labor-intensive effort, but frameworks like STIGs and RMF help manage the complexity by giving structured guidance on what to do.

Finally, **documentation** is a part of secure engineering. Every system should have an up-to-date network diagram, data flow diagram, and security plan documenting its components and controls. This not only helps in audits/ATOs but is invaluable in troubleshooting and responding to incidents (knowing what connects to what, where data resides, etc.). As the saying goes, *"security by obscurity is not a strategy"* – clarity and knowledge of your environment is foundational.

In summary, secure systems engineering means **baking security into every layer and every stage of a system's lifecycle**. By doing so in local on-prem environments and extending those principles to cloud or hybrid setups, agencies create strong, defensible infrastructure. Coupled with user-focused measures (like training and access control) and advanced topics like PQC and Zero Trust discussed earlier, this forms a holistic shield around classified and sensitive work.

16

Secure Communication and Collaboration

Communication – the exchange of information – is the lifeblood of government operations. Whether it's a phone call, an email, a videoconference, or data sharing between systems, ensuring these communications remain **confidential, authenticated, and untampered** is crucial when sensitive information is involved. Secure communication isn't just about technology; it's also about usage policies and user behavior (avoiding insecure channels). Here, we cover how to establish secure connections and collaboration mechanisms, ranging from interpersonal communication to machine-to-machine data exchange.

Encrypted Voice and Video Communications

- **Secure Telephony (Voice):** Traditionally, classified voice communication was done via secure telephones (e.g., the STE – Secure Terminal Equipment, or its successors).

These devices encrypt calls end-to-end using NSA-approved algorithms when both parties use secure phones. Now, VoIP has become common, and NSA certifies some VoIP systems for classified use (e.g., VoSIP on SIPRNet for Secret-level voice). For sensitive but unclassified, agencies might use encrypted VoIP apps or services – even commercial solutions like Signal or Skype with strong encryption, but preferably FedRAMP-approved ones or agency-hosted solutions for full control. **Any voice discussing sensitive topics should be encrypted;** if a secure line isn't available, then it's often advised to wait rather than use an open line. Secure mobile devices (like NSA's CSfC program smartphones) allow secure voice over cellular or Wi-Fi by running voice encryption apps that create a VPN back to a government network.

- **Secure Video Conferencing:** Video calls add complexity (video, audio, possibly screen sharing), but similar principles apply. There are high-secure video systems (e.g., JWICS has systems for Top Secret video meetings among IC members). At lower levels, especially post-COVID, agencies widely adopted tools like Zoom for Government, WebEx for Government, Microsoft Teams, etc., which are FedRAMP authorized and offer end-to-end encryption options. For instance, NSA published guidance on configuring ZoomGov for sensitive unclassified discussions (like enabling waiting rooms, using passcodes, not sharing links publicly, etc.). For classified video conferencing, dedicated hardware in SCIFs, like secure VTC units, are used with networks like SIPRNet or JWICS handling the encrypted transport. **Physical security** is also important: even if the video feed is encrypted, one must conduct classified

video calls from within a controlled area (so no uncleared persons overhear or see something on screen).

- **Voicemail and Recording:** Secure systems usually disable voicemail or record features for classified conversations, because recording introduces stored data that must then be protected. On unclassified but sensitive systems, if meetings are recorded, they need to be stored securely (and possibly with access limited). Many agencies treat recordings or transcripts of meetings as sensitive data that needs the same protections as an equivalent document.

Secure Email and Messaging

- **Email Encryption (S/MIME, PGP):** Government personnel often have the capability to send encrypted emails. The most common method in US Govt is using **S/MIME** with the certificates on the user's PIV card. For instance, if Alice at DHS wants to email Bob at DOJ something sensitive, she can use Outlook with S/MIME to encrypt the email such that only Bob's private key (on his PIV card) can decrypt it. All federal employees have S/MIME certs via the Federal PKI, and the global address list in agencies often contains each person's public cert to facilitate this. Using S/MIME for sensitive emails (and signing them for integrity) is a best practice – in fact, EO 14028 required federal agencies to move towards **mandatory encryption for email in transit** (like ensuring SMTP uses STARTTLS) and ideally end-to-end for sensitive content. On classified networks, email is protected by the network's encryption (e.g., SIPRNet emails are encrypted by the network layer and often also signed/encrypted via NSA mechanisms).

- **Controlled Email Infrastructure:** Agencies run their own email servers or use government cloud (like Office 365 GCC High or DoD 365). These environments enforce policies: blocking auto-forward to external accounts, scanning attachments for malware, using DLP to prevent certain data from being emailed out (e.g., SSNs or "Classified" markings triggering a send block). For highly sensitive projects, sometimes separate email accounts or systems are used with limited user lists (compartmentalization even in email).
- **Instant Messaging/Chat:** Quick communication via chat is common (tools like Teams, Slack, or on classified nets, tools like JWICS chat). These need to be secure as well. Most enterprise chat systems now encrypt data in transit and at rest. For classified, chat is usually within the closed network with added encryption (like OTE – one-time pad encryption if extremely sensitive, but that's rare). Policies need to cover chat content – e.g., not pasting classified data into an unclassified chat. Some agencies deploy their own XMPP or Matrix chat servers for internal use to ensure full control. It's important to note that even metadata like who is chatting with whom might be sensitive (traffic analysis concerns), so secure comms consider protecting that too (some systems offer off-the-record messaging, though that conflicts with government record-keeping requirements – a balance must be struck).
- **Removable Media and File Transfer:** Often communication isn't person-to-person but system-to-system or person-to-system (like uploading a file). Secure **file transfer** protocols (SFTP, HTTPS, or dedicated secure file share portals) should be used instead of unencrypted FTP or,

worse, personal cloud drives. Many agencies provide a secure file drop service for large files (with encryption and access control). On classified systems, moving files out is extremely controlled – usually requiring a review and use of a sanctioned transfer method via cross-domain guards or courier on encrypted USB if digital.

- **Personal Devices and Communication:** A repeated theme – **don't use personal email for work**, don't discuss work on personal phones or social media. To enforce this, agencies sometimes implement DLP that detects if someone tries to email their personal account with work data. Mobile Device Management (MDM) can also separate work data on mobile phones so that, for example, work email cannot forward to a personal app. Training reminds employees that even unclassified work info often shouldn't be on personal platforms due to sensitivity or at least Privacy Act considerations.
- **Physical Mail and Delivery:** Though much is digital, sometimes sensitive info is sent physically (couriers, secure mail). Classified materials are sent via cleared couriers or via diplomatic pouches, etc., and double-wrapped. This is outside our main scope, but it's a communication method that has its own set of secure handling rules (classified mail cannot be sent through regular postal services, for instance).

Inter-Agency and Coalition Communication

Government work often involves sharing sensitive info with other agencies, state/local governments, or allied nations. Secure communication extends to these external partners:

- **Cross-Agency Collaboration:** The federal government has networks like **NSAnet or Intelink** for intel sharing, and platforms like **MAX.gov** or **HSIN** (Homeland Security Information Network) for inter-agency data at lower classifications. These are secured enclaves where multiple orgs have accounts and share info with role-based access. When collaborating, say, on a law enforcement case, they might use an FBI system with appropriate accounts for other agency personnel, rather than emailing files around. This ensures data stays within controlled systems with proper auditing.
- **State and Local Partners:** Programs like MS-ISAC (Multi-State Information Sharing & Analysis Center) facilitate sharing cyber threat info securely with states. Also, systems like FBI's CJIS (Criminal Justice Information Services) network connect state police to federal data – CJIS has its own stringent security policy that state/local agencies must follow (e.g., FBI requires MFA, certain encryption, etc., for any system accessing CJIS data).
- **International (Coalition) Sharing:** Within NATO or Five Eyes, there are secure systems (like **STONEGHOST** for Secret-level info among Five Eyes). Bilateral channels exist too (e.g., a secure voice line between Pentagon and a counterpart). When the U.S. shares classified info with allies, it often gets re-marked with joint classification levels (e.g., SECRET//REL TO USA and [Country]) and goes over approved secure channels. Coalition networks in military operations (e.g., a Combined Enterprise Regional Information Exchange System – CENTRIXS) are set up to allow US and coalition partners to share relevant intel and comms without exposing everything from each side.

These are engineered with separate authentication and guard systems to prevent, say, a coalition member from accessing U.S.-only info.

- **Public Communication of Sensitive Info:** Sometimes, agencies need to communicate with the public or media about sensitive matters. They use secure methods to send embargoed press releases or handle data sharing. Notably, if something can't be shared securely, they may opt for in-person only briefings (like showing documents in a reading room instead of sending files). This is part of risk-based decisions on communication.

Human Factors and Operational Security (OPSEC)

No discussion of secure comms is complete without addressing the **human aspect**:

- **Training on Secure Communication:** Employees are trained on what mediums are approved for what classification. For instance, "don't talk about classified in an open area or on your personal phone" is drilled in. Also, "check who you're talking to" – social engineers might call pretending to be someone. Always verify identities for any non-routine communications (e.g., call them back on a known number).
- **OPSEC in Conversations:** Even in secure buildings, sensitive conversations should occur behind closed doors or in designated areas. The government often has posters reminding "**Loose lips sink ships**" – a WWII slogan still relevant. For example, you wouldn't talk about a sensitive operation in the cafeteria, even if mostly cleared folks are

around, because someone who isn't cleared might overhear.

- **Use of Codewords or Nicknames:** Sometimes within secure channels, additional OPSEC is practiced by using codewords or minimizing details. E.g., referring to a project by a codename even in classified discussion can limit damage if overheard. This is more relevant in military/intel context.
- **Emerging Tech – Quantum Communications:** Looking ahead, the government and industry are also exploring **quantum key distribution (QKD)** as a means of ultra-secure communication. QKD allows two parties to generate a shared encryption key with the property that any eavesdropping attempt is detectable (thanks to quantum physics). While not widely deployed, some initial use in government or critical infrastructure might occur for the highest security requirements (perhaps connecting major data centers or leadership comms). It's complementary to post-quantum crypto – QKD secures the key exchange channel, but it's expensive and distance-limited.

To summarize, secure communication and connections are achieved by **applying encryption and strong authentication at every layer**, using approved tools and networks, and by **ensuring users follow protocols not to bypass these measures**. From picking up a secure phone for a classified call, to using encrypted email for sensitive memos, to configuring APIs to use TLS with client certificates – it's a culture and practice of **always using the most secure method available that is appropriate for the information at hand**. In government, where the stakes are high, this often means dedicated infrastructure and rigorous rules. The technologies

and frameworks we've covered (Zero Trust, ICAM, PQC, etc.) all converge to enable secure communications, but it takes vigilance to use them correctly and consistently. By doing so, agencies keep adversaries in the dark and sensitive information in the right hands only.

17

Hardware and Supply Chain Security

The security of government systems is not only determined by how they are configured and used, but also by **how they are built and from where they come**. Adversaries may target the supply chain – the sequence of designing, manufacturing, shipping, and maintaining hardware and software – to insert vulnerabilities or malicious components. For example, a corrupted firmware in a network router or a backdoor in a software library could undermine an agency's defenses. Therefore, a comprehensive security program addresses **hardware security and supply chain risk management (SCRM)** to achieve a level of **zero trust in the supply chain** – essentially not implicitly trusting any component until verified and continually assessed.

Hardware Supply Chain

- **Trusted Foundries and Vendors:** For the most sensitive hardware (cryptographic modules, military chips, etc.), the U.S. has programs to use **trusted foundries** – manufactur-

ers that are vetted and often on U.S. soil. For commercial off-the-shelf (COTS) hardware, agencies rely on approved vendor lists. The government maintains lists of banned hardware (as per the NDAA Section 889, equipment from certain Chinese companies can't be used). Procurement rules enforce checking those lists. Agencies also look at vendor reputation: Do they follow secure development? Have they had incidents of supply chain compromise? Supply chain risk assessments are sometimes done before big purchases.

- **Chain of Custody:** When hardware is in transit, there's risk of interdiction (e.g., an adversary intercepting a shipment to implant something). High-security deliveries might use tamper-evident seals on packages and authorized couriers. Upon receipt, hardware can be inspected for signs of tampering (some agencies do X-ray or visual inspection of circuit boards for unrecognized chips). **Secure storage** is used for spare equipment – not just lying around where someone could swap it.
- **Hardware Authenticity Verification:** Techniques like verifying hardware serial numbers and cryptographic attestations (if the hardware supports it) help ensure a device is genuine. Some hardware (like Cisco routers) have built-in trust anchors that can prove to a management system that they're running untampered firmware from the vendor. Utilizing those features (Cisco's SUDI – Secure Unique Device Identity, for instance) can raise assurance that hardware isn't counterfeit or modified.
- **Firmware Security:** The supply chain extends to firmware and microcode. Agencies keep firmware updated (to patch flaws) but also ensure they obtain firmware from

official sources (manufacturer websites, etc.) and verify digital signatures when applying updates. There's also the concept of **Firmware Integrity Monitoring** – some EDR solutions now look at BIOS/UEFI images to ensure they match expected hashes, detecting if malware implanted a backdoor in firmware. Given that firmware runs at a very privileged level, it's a juicy target, so securing it via signed updates and verification is key.

- **Physical Element of Hardware Security:** Protecting hardware from physical interference includes controlling access to sensitive hardware (only authorized technicians can service it) and possibly using anti-tamper locks or coatings. For example, critical servers might have locks on the rack and chassis intrusion detection. Some agencies apply tamper-evident tape or seals on ports or screws of critical devices (so if someone tries to open it, the seal breaks). While perhaps old-school, these methods can deter or at least flag hardware tampering attempts by insiders or during transport.

18

Software Supply Chain and Third-Party Components

Modern systems involve a lot of third-party software – open source libraries, commercial products, etc. Supply chain security for software involves:

- **Source Integrity:** Using trusted repositories for open-source components (like pulling from vendor-signed packages or verifying hashes against known good). Tools like dependency checkers, SBOMs, and code signing help ensure the code running is exactly what's expected. The SolarWinds incident, where an update server was compromised to deliver trojaned updates, highlights the need for verifying the integrity of software updates. Agencies now often require vendors to provide signatures for updates and have processes to validate them before deployment (many use an **update whitelist** – only allowing software from certain update servers or code-signing CAs).
- **Secure Build and Delivery:** If an agency builds software in-house, the build pipeline must be secured (to prevent

injection of malicious code). That might mean isolated build servers, use of reproducible builds, and signing the final binaries. If using cloud CI/CD, ensuring it's hardened and credentials to repositories are protected. After build, distribution should be via secure channels (internally, perhaps via an artifact repository that is access-controlled).

- **Software Bill of Materials (SBOM):** By maintaining an SBOM, agencies know what components (and versions) are in their software. This is invaluable when a new vulnerability (like Log4j in 2021) is announced – they can quickly search SBOMs to find all systems that have that component. The EO 14028 push means agencies are asking for SBOMs from vendors for critical software, and tools are emerging to automatically ingest SBOMs and continuously watch for newly disclosed component vulns (some tie into NIST's National Vulnerability Database to alert if a CVE matches a component in an SBOM).
- **Security Testing of Third-Party Software:** Before deploying a third-party application, agencies may conduct their own security assessment – code review if source is available, or at least dynamic testing. Especially for anything touching sensitive data, one might put it in a sandbox environment and fuzz it or run SAST/DAST tools. There's also the concept of **Zero Trust for apps** – not fully trusting an app just because it's approved. This means running it with least privileges, maybe containerizing it, and monitoring its behavior.
- **Vendor Attestations and Audits:** As noted, OMB policy now requires vendors to **attest** they follow secure development practices. Over time, we might see third-

party audits or certifications of vendor security (similar to how cloud providers get FedRAMP certification). For now, agencies might include contract clauses giving them the right to review certain vendor security practices or requiring compliance with standards like ISO 27001 or the new **Secure Software Development Framework (NIST SSDF)**. This pushes the supply chain to be accountable.

- **Open Source Use:** The government uses open source a lot (some agencies even have open source first policies). But with that comes risk of malicious packages (typosquatting attacks on package managers) or abandoned projects with no maintenance. Agencies mitigate this by using internal package repositories (hosting vetted copies of open source libraries and only allowing developers to use those), and tracking open source communities for any compromise events. DHS's CISA even launched an initiative for **open-source supply chain security** (looking at things like signing commits, etc., in major projects). A practical step for agencies is to subscribe to vulnerability feeds for open source components (like using GitHub's Dependabot alerts or similar) and patch promptly.

19

Monitoring and Responding to Supply Chain Threats

Supply chain security is an ongoing process:

- **Threat Intelligence:** Stay informed about supply chain threats. For instance, if an adversary is known to target a certain brand of equipment or a certain software supplier, that info (often via FBI or CISA alerts) should prompt action (like extra scrutiny on that brand or supplier). The Intelligence Community shares relevant classified intel with cleared industry and agencies under programs like the NSA's Cybersecurity Collaboration Center, which might tip off that "X country may attempt supply chain compromises via Y distributor."
- **Incident Response for Supply Chain Issues:** If a compromised component is discovered (e.g., it's found a batch of security cameras has a hidden backdoor communicating to overseas servers), agencies need to be ready to rip-and-replace or apply mitigations quickly. That means having an

inventory of where those are, possibly network isolating them until fixed, etc. As an example, when the SolarWinds backdoor was uncovered, agencies had to within days identify all instances of that software, disconnect or patch them, and hunt for any signs the backdoor had been exploited.

- **Continuous Evaluation of Suppliers:** Supply chain risk management suggests periodically reviewing suppliers' security posture. If a vendor has repeated security issues, an agency might phase them out. Conversely, agencies might give preference in contracts to vendors with a strong security track record. The federal government has discussed ideas like "**Software Liability**" or secure-by-design principles, which, if implemented, could hold suppliers legally responsible for egregious security flaws. While not yet concrete, it shows a direction where security expectations on suppliers will only increase.

20

Physical Supply Chain & Counterfeit Prevention

Hardware security includes making sure you're using genuine, quality parts:

- **Counterfeit Hardware:** Apart from malicious backdoors, there's risk of cheap counterfeit parts (which might fail or have unknown origin firmware). Agencies combat this by sourcing from authorized resellers and using programs like GIDEP (Government-Industry Data Exchange Program) which shares info on suspect counterfeit parts. If a random vendor offers Cisco routers at 80% discount, that's a red flag – they could be counterfeit running modified OS. Thus procurement officials are trained to spot deals "too good to be true" and ensure traceability to OEMs.
- **Maintenance and Updates:** When hardware is sent for maintenance, there's risk of tampering. Some agencies have policies forbidding sending back certain sensitive hardware to the vendor

for RMA (they'll destroy it instead and buy new, to avoid a hard drive with secrets going to a factory). Or if a tech comes on-site, they are escorted. Also, after maintenance, one might re-verify the hardware/firmware hasn't changed unexpectedly.

In essence, **supply chain security extends trust boundaries** – recognizing that security isn't just what you do in-house, but also what happens upstream (with suppliers) and downstream (with maintenance, disposal). The key to supply chain security is a combination of **procurement controls, technical verifications, and continuous vigilance**. It's implementing the old adage "trust, but verify" to every component that enters your environment. By doing so, agencies aim to prevent "trojan horses" from bypassing their fortress walls – because a slickly engineered internal security system can be undermined if the hardware itself comes pre-compromised or the software libraries include hidden spies. The U.S. government's focus via regulations and frameworks (like NIST 800-161 and Section 4 of EO 14028) shows that supply chain security is now a first-class element of cybersecurity programs.

21

Personnel Security and Insider Threats

While much of cybersecurity focuses on technology, history shows that **people** – whether through errors, coercion, or malice – can pose some of the greatest risks to safeguarding sensitive information. Governments have long recognized the "insider threat," exemplified by infamous cases (spies, leakers, or simply negligent employees), and have developed extensive **personnel security** measures. These measures work in tandem with technical controls to create a comprehensive security posture. We will discuss how to ensure the **people** who have access to classified or sensitive work are trustworthy, well-trained, and monitored in a way that deters and detects risky behavior, while also fostering a culture of security.

The Security Clearance System

- **Background Investigations:** To gain access to classified information, individuals must undergo a thorough **background investigation** and adjudication process to

receive a security clearance at the required level (Confidential, Secret, or Top Secret, often with additional caveats like SCI eligibility). These investigations, conducted by agencies like OPM or DoD's DCSA, comb through an individual's criminal history, credit reports, references, foreign contacts, etc. The aim is to identify any **red flags**: criminal behavior, signs of unreliability, vulnerability to blackmail (e.g., large unexplained debts), allegiance issues, or personal conduct issues (like drug abuse) that might make them a security risk. The criteria are spelled out in the **Adjudicative Guidelines** (covering 13 areas such as Finances, Foreign Influence, Emotional Disorders, etc.). By ensuring only vetted individuals are granted access, the government mitigates the risk of an adversary simply placing an agent in a sensitive role or an unstable person having access to secrets.

- **Polygraphs and Additional Screening:** Certain particularly sensitive positions (e.g., CIA operations, NSA analysts, nuclear weapons personnel) may require **polygraph examinations** as part of the clearance process or periodic evaluations. Polygraphs, while controversial, are used to try to catch undisclosed behavior or information (like hidden drug use or undisclosed foreign contacts) and deter people from lying on their clearance forms. Some agencies also do psychological evaluations or mental health screening, not to discriminate, but to assess if an individual's mental/emotional state might pose a risk (e.g., extreme stress that could lead to a breakdown or susceptibility to coercion). Additionally, a **Continuous Evaluation (CE)** program is now supplementing periodic reinvestigations: automated checks of cleared individuals'

records (criminal, credit, etc.) happen frequently, so if, say, someone is arrested or falls into big debt, security officers are alerted in near-real-time.

- **Need-to-Know Enforcement:** As mentioned earlier, beyond clearance level, *need-to-know* is enforced through **compartmentalization**. Special Access Programs (SAPs) and SCI control systems mean that even a Top Secret-cleared person can't just roam through all Top Secret information – they must be *"read in"* to each program. Technically, this is enforced by separate networks or enclaves and by data labeling. Culturally, it's enforced by reminding people not to share info with colleagues unless they're sure the colleague is cleared *and* officially authorized for that info. Many a friendly conversation in the hallway starts with "Hey, are you read into Program X? No? Okay, never mind."
- **Security Briefings and Debriefings:** When someone gets a clearance, they must attend a security briefing where they acknowledge their responsibility (sign NDAs like SF312 Classified Information Nondisclosure Agreement) and learn the rules for handling classified information. Equally important, when someone leaves a position or a program (or leaves government service), they get a debriefing – reminding them they are **obligated for life** to protect the information they were exposed to, and retrieving any badges or materials. They also sign acknowledgments of continuing responsibility. This doesn't technically prevent someone from leaking secrets after they quit (see Snowden or many others), but it's a legal instrument enabling prosecution if they do, and a moment to reinforce the gravity of their obligations.

Training and Awareness

- **Regular Security Training:** All personnel, cleared or not, receive routine **cybersecurity and information security awareness training** (often annually). This training covers topics like identifying phishing attempts, proper use of passwords/PIV cards, how to spot social engineering, physical security for documents, and what to do if one suspects a security incident or compromise. It's often interactive, with scenario examples: e.g., "You find a USB in the parking lot – what do you do?" (Correct answer: don't plug it in; turn it over to security). For cleared individuals, there's additional training on handling classified info (like classification marking, the do's and don'ts of discussing classified, how to destroy classified media, etc.). There's also usually a "**Insider Threat Awareness**" module – teaching employees to recognize signs that a coworker might be up to something (unusual hours, attempts to access unrelated info, sudden affluence, or expressions of resentment). The idea is to create many eyes that can catch an insider before they do damage.
- **Specialized Training:** Certain roles get extra training – e.g., IT administrators get training on how to avoid being manipulated (like someone calling pretending to be a user asking for password reset). Executives receive training because they're high-value targets (things like being careful with what they say in public or at conferences). Also, individuals traveling abroad get briefings about risks (hotel room may be bugged, foreign intel may approach them under friendly pretenses, their devices could be targeted – often they're given clean loaner devices for travel and told

not to discuss work abroad).

- **Reporting Requirements:** Personnel are required to self-report certain things: contact with foreign nationals (in some cases), any attempts by strangers to obtain info, any security incidents they cause (like a misplaced document), and personal issues like an arrest or financial trouble. This can feel intrusive, but it's to allow security officers to evaluate if support or action is needed (sometimes help – like counseling or financial counseling – is offered, because an employee under stress is a risk and helping them mitigates that risk). Co-workers are likewise encouraged (or required) to report suspicious behavior or incidents. The environment tries to balance **vigilance with trust** – you don't want a Stasi-like snitch culture, but you do want people to speak up if something is clearly off. Many agencies foster this by messaging that reporting isn't about getting someone in trouble, it's about protecting everyone and getting ahead of issues.

Insider Threat Programs and Monitoring

Post-WikiLeaks and Snowden, the government greatly enhanced **Insider Threat Programs**:

- **User Activity Monitoring:** On classified networks, agencies implemented tools that monitor user actions at a fine level – for example, systems can flag if a user is downloading an unusually large amount of data, or plugging in an unauthorized device. There are even tools that can detect if a user is copy-pasting chunks of documents or taking screenshots. Some use heuristics or AI to detect anomalous

patterns per user. This is delicate since it can infringe on privacy; however, users of classified systems have reduced expectations of privacy (they consent to monitoring as a condition of access). The monitoring data is handled carefully and only reviewed by authorized insider threat analysts.

- **Data Labeling and Preventing Unauthorized Access:** All classified files are (or should be) labeled with classification and dissemination controls. Systems like **Rights Management Services** can enforce who can open a document or forward it. For example, if a document is labeled NOFORN, the system can prevent sending it to a foreign partner email. Modern implementations might tie into identity attributes – e.g., only people with a certain role or who are read into a compartment can decrypt a file from that compartment. Insider threat mitigation includes using such technology to **technically enforce need-to-know**, so even if someone tries to hoard data they aren't entitled to, they hit barriers.
- **Behavioral Analytics:** Insider Threat Programs compile data not just from IT but also from HR and security files: e.g., is an employee disgruntled (bad performance review, conflict with colleagues), did they recently have a personal crisis, are they working odd hours (could be dedication… or could be surreptitious activity). By correlating technical indicators with human factors, programs try to identify at-risk individuals for intervention. If someone is flagged, they might be quietly investigated, or their access might be limited pending a review.
- **Sanctions and Deterrence:** Knowing that monitoring is in place and that there's a high chance of getting caught can

deter insiders. But if someone is caught doing something (even minor, like looking at files they aren't supposed to), agencies often take administrative action – counseling, reprimand, or clearance suspension – to reinforce seriousness. For willful misconduct, they can be fired and prosecuted. The Justice Dept has prosecuted numerous leakers in the past decade under the Espionage Act even when they weren't spies but people giving info to media. The high penalties (potential prison) serve as deterrence.

- **Positive Work Environment:** Interestingly, one of the best insider threat mitigations is maintaining a positive workplace and mission-focused culture. Many insider betrayals were motivated by disgruntlement or ethical outrage. Agencies try (or should try) to address employee concerns, provide avenues for whistleblowing through proper channels, and build trust so that employees won't feel the need to take matters into their own hands in destructive ways. A content, engaged workforce is less likely to go rogue. This touches on leadership, management, and even how security policies are enforced (e.g., not too oppressive without explanation, so they don't breed resentment).

Physical Security & Personnel

Physical and personnel security overlap significantly:

- **Facility Clearances and Badging:** Only cleared personnel (or escorted visitors) can enter secure facilities where classified work is done. Physical access systems (badge readers, biometric scanners) ensure that only those with

the right clearance level and need are allowed. Many SCIFs have *security officers* or receptionists who check badges and IDs. There's also the practice of wearing color-coded badges indicating clearance level or status (green stripe might mean TS/SCI, red might mean uncleared visitor, etc.) so one can visually spot if someone is out-of-place.

- **Two-Person Rule:** In certain high-risk operations (like handling nuclear launch codes or certain cryptographic keys), a **two-person rule** is enforced: no single person can perform the critical action or be left alone with the sensitive materials. This prevents an insider from acting unilaterally (you'd need collusion, which is statistically less likely). Even in day-to-day, having at least two people present in a secure lab can prevent mishandling or quickly detect if someone is doing something suspicious.
- **Periodic Reinvestigations and Polygraphs:** Historically TS clearances were reinvestigated every 5 years, Secret every 10. With **Continuous Vetting**, that schedule is changing (many checks are ongoing, with perhaps a formal review at longer intervals). Polygraph programs sometimes also re-polygraph people every 5 years or randomly. This isn't foolproof (people can lie or not get caught by poly), but it's another tool that might deter insiders who know they'll be asked regularly about unauthorized disclosures or foreign contacts.
- **Exit Procedures:** When someone with knowledge leaves (retires, quits), they get a debrief and their accounts are disabled immediately. If someone is fired or resigns under bad terms, sometimes they're escorted out immediately and not allowed to touch their computer. This is to prevent a disgruntled person from planting logic bombs or stealing

data on their way out. Some spectacular insider incidents happened at time of resignation. So HR and security coordinate closely on personnel departures to mitigate that.

In conclusion, **personnel security and insider threat mitigation form the human-centric layer of defense**. All the encryption and firewalls in the world can be undone if a trusted person decides to leak or if an untrustworthy person infiltrates. Therefore, the government expends great effort to **vet, educate, monitor, and when necessary, discipline** the people in the system. It's a delicate balance protecting civil liberties of employees and fostering a positive workplace, while also maintaining vigilance for the rare (but high-impact) malicious insider or the more common careless action. The key is creating a culture where security is part of everyone's job and where help is available (e.g., for personal issues or questions) so that problems can be addressed openly rather than festering and leading to a breach. By combining these personnel measures with the technical controls discussed elsewhere, agencies approach the goal of "**zero trust" not just in networks but in all aspects – verify trustworthiness and actions of even insiders** as much as practical, to protect national security information.

22

Applications That Never Shut Up

Why Software Constantly Leaks Data and Why Secure Systems Must Be Ruthless

A secure system is not compromised only by hackers. It is compromised by **chatty software**. Modern applications leak data continuously, quietly, and often legally. They phone home, sync state, check licenses, fetch ads, update models, log telemetry, prefetch content, refresh tokens, and negotiate background connections even when you believe they are idle.

Security does not fail in dramatic moments. It fails in the background.

If a system processes sensitive information, **every installed application is a potential transmitter**, whether or not it was designed with malicious intent.

1. The uncomfortable truth: most applications are not designed to be silent

Modern software is built for connectivity, analytics, monetization, and user engagement. Silence is not a design goal.

Typical background behaviors include:

- Telemetry and analytics uploads
- Crash and performance reporting
- License validation and entitlement checks
- Cloud sync and backup polling
- Content preloading and recommendation refresh
- Feature flag checks
- A/B testing configuration fetches
- Ad network signaling
- Model updates or remote inference calls
- Time-based "heartbeat" connections

These behaviors often occur:

- At boot
- On network reconnect
- On idle
- On wake from sleep
- On clock tick
- On user context change
- On resource threshold changes

From a security standpoint, this means **you do not control when data leaves the system** unless you explicitly engineer for that control.

2. Leakage is not only about content

Most people think of leakage as "files being sent out." That is an incomplete model.

Applications leak **metadata**, which is often more valuable than content:

- Timing patterns
- Usage frequency
- Feature access
- Hardware identifiers
- OS configuration
- Installed software list
- Locale and language
- Power state changes
- Network topology hints
- Unique identifiers that persist across reinstalls

Metadata allows adversaries to:

- Fingerprint systems
- Infer workflows
- Identify sensitive operational windows
- Correlate users across environments
- Detect when sensitive work is happening

You can encrypt content perfectly and still leak everything else.

3. Backdoors are not always malicious

When people hear "backdoor," they imagine hidden hostile code. In practice, many backdoors are **intentional features** that violate security expectations.

Examples:

- Remote debugging hooks
- Vendor support access
- Auto-update frameworks
- Embedded web servers
- Cloud management agents
- Remote configuration toggles
- Undocumented APIs
- Hard-coded trust anchors
- Third-party SDKs with network access

From a security perspective, intent does not matter. Capability does.

If code can transmit data or accept remote instructions, it is a backdoor relative to a high-assurance system.

4. The false comfort of "closed source" and "open source"

Neither closed-source nor open-source software is automatically safe.

- Closed source hides behavior you cannot audit.
- Open source exposes code, but not build pipelines, dependencies, or runtime configuration.

Common failure modes:

- Open-source app with proprietary telemetry SDK
- Closed-source app using encrypted channels with undocumented payloads
- "Audited" software that updates silently
- Signed binaries that change behavior post-install via remote configuration

Security requires **runtime control**, not philosophical alignment.

5. The background network problem

Most operating systems encourage applications to manage their own networking. This creates a critical issue:

You do not know when traffic is expected and when it is not.

Background transmissions may occur:

- Outside working hours
- During sensitive operations
- While air-gapped assumptions are violated
- Through unexpected interfaces (Wi-Fi, Bluetooth, cellular, loopback tunneling)

If an application can initiate outbound connections freely, **you have lost temporal control** over data flow.

Temporal control is as important as content control.

6. Why "I was not using it" is irrelevant

An application does not need to be visible to be active.
Background processes can:

- Wake periodically
- React to system events
- Respond to network broadcasts
- React to hardware interrupts
- Process cached queues
- Flush logs on reconnect

From a defensive standpoint:

- Idle does not mean inert
- Minimized does not mean inactive
- Closed UI does not mean terminated

Security systems must assume **asynchronous behavior**.

7. Secure systems require software minimalism

High-assurance systems are intentionally boring.
Core principle:

> *Every additional application increases the attack surface, the leakage surface, and the unpredictability of system behavior.*

Secure systems follow these rules:

- Only install software with a clear operational necessity
- No "just in case" tools
- No convenience applications
- No personal software
- No auto-updaters without strict control
- No background services without justification

If software cannot justify its presence in one sentence, it does not belong.

8. Trust is not binary, it is contextual

Software may be safe in one environment and unacceptable in another.

Examples:

- A cloud IDE is fine on a development laptop, unacceptable on a classified system
- A browser extension is fine for personal use, unacceptable on a compliance workstation
- A messaging app is fine for logistics, unacceptable for design review
- AI assistants may be fine for brainstorming, unacceptable for sensitive analysis

Security is about **contextual trust boundaries**, not absolute trust.

9. The myth of "offline mode"

Many applications claim an offline mode. Few truly respect it.

Offline mode often means:

- No user-facing network features
- But background retries still queue
- Logs still accumulate
- DNS still resolves
- Certificates still refresh
- Time sync still attempts
- Local network discovery still occurs

True offline behavior must be **enforced at the system boundary**, not delegated to application goodwill.

10. Enforcing silence: architectural controls

To prevent unwanted transmission, control must exist outside the application.

Effective controls include:

- Default-deny outbound networking
- Per-application network allowlists
- Time-based network enablement
- Physical network isolation
- Interface-level shutdown (Wi-Fi, Bluetooth, radios)
- Mandatory access control policies
- Process-level network namespaces
- Audited firewall rules with logging

If an application cannot transmit without explicit permission, it becomes predictable.

Predictability is security.

11. Supply chain and update risks

Updates are one of the most dangerous moments in a system's lifecycle.

Risks:

- New telemetry introduced post-install
- New dependencies added silently
- Behavior changes without notice
- Expanded permissions
- Remote feature toggles activated later

Secure systems:

- Do not auto-update blindly
- Stage updates in test environments
- Verify behavior after update
- Lock versions during sensitive operations
- Treat updates as configuration changes, not maintenance

An update is a code injection event. Treat it as such.

12. Psychological traps that break secure discipline

Even experienced engineers fall into these traps:

- "It is just a small utility"

- "Everyone uses this"
- "It is reputable"
- "I trust the vendor"
- "I will uninstall it later"
- "It does not ask for permissions"
- "It is only for convenience"

Security failures are rarely technical. They are behavioral.

13. Secure systems must assume software is guilty until proven silent

The correct default stance is not paranoia. It is engineering realism.

Assume:

- Software will transmit data if allowed
- Software will change behavior over time
- Software will inherit risk from dependencies
- Software will react to events you did not anticipate

Therefore:

- Do not install unless necessary
- Do not allow network unless required
- Do not allow background execution unless justified
- Do not allow updates unless controlled
- Do not mix trust domains on the same system

14. The core rule

If you are using a system for security-sensitive work:
Do not install applications unless you have verified that:

- You understand exactly what data they transmit
- You control when they can transmit
- You can prevent transmission entirely
- You can audit or observe their behavior
- You can remove them completely if needed

Anything less is hope. Hope is not a security strategy.

Closing thought

A secure system is not powerful because of what it has.
It is powerful because of what it refuses to run.
The quietest machine is the hardest to betray.

23

Continuous Monitoring, Incident Response, and Improvement

Security is not a "set and forget" endeavor – it's a continuous process. Even with all the preventative measures discussed, incidents can (and likely will) still happen, and new vulnerabilities or threats will emerge. Thus, a resilient cybersecurity program must have strong **continuous monitoring** to detect issues promptly, a robust **incident response** capability to handle incidents when they occur, and a commitment to **continuous improvement** – learning from incidents and adapting strategies, as well as keeping up with evolving best practices and threat trends.

Continuous Monitoring and Detection

- **SIEM and SOC Operations:** Most agencies operate a Security Operations Center (SOC) or leverage a shared SOC (sometimes via DHS/CISA for smaller agencies) where analysts watch for alerts. They use a **SIEM (Security Information and Event Management)** system that ingests logs

from across the enterprise: system logs, firewall logs, IDS alerts, cloud service logs, etc. The SIEM correlates events (e.g., a VPN login from an unusual country + subsequent large data access might trigger a higher alert). Continuous monitoring means these systems run 24/7, with use cases and rules continually updated based on intel and past incidents. For example, after seeing a type of phishing, a detection rule might be added to flag if any user visits the URL that was used in that phishing campaign.

- **Automated Endpoint Monitoring:** Tools like EDR (Endpoint Detection & Response) and NDR (Network Detection & Response) provide real-time monitoring at the endpoint and network level respectively. They often utilize machine learning to spot anomalies like a process spawning a command prompt and making network connections (could be malware) or a host scanning multiple other hosts (could be lateral movement). These can either auto-contain a threat (e.g., isolate a machine from network if suspected infected) or at least alert SOC. Continuous monitoring configures these tools with policies that align to the environment (to reduce false positives but catch true threats).
- **Threat Hunting:** Beyond reactive alert handling, many SOCs do **proactive threat hunting**. They hypothesize about possible undetected threats and then search the environment for signs. For instance, "If an attacker had compromised an admin account, what traces might that leave? Let's search logs for any unusual admin actions at odd hours." Threat hunting often uses threat intelligence (like indicators of compromise – file hashes, IP addresses, malicious domain names) to sweep through data and see if any match appears. This continuous activity helps catch

stealthy attackers that automated tools didn't flag.

- **Anomaly Detection and User Behavior Analytics:** Some agencies implement **UBA/UEBA (User and Entity Behavior Analytics)** systems. These baseline normal behavior for users and entities (devices, service accounts) and then raise alerts on deviations. For example, if Bob typically accesses 3 systems and 10 files a day, and suddenly accesses 30 systems and 1000 files, that's flagged. Or if a printer starts sending outbound network traffic (which it normally never does), that's flagged as an anomaly. UEBA can be a component of zero trust, feeding into access decisions as well as monitoring.
- **Continuous Vulnerability Scanning:** Part of continuous monitoring is also scanning for vulnerabilities regularly. Tools scan networks for open ports, outdated software, misconfigurations. The DHS CDM program provides some sensors to agencies for this purpose. **Web app scanners** also run regularly on internet-facing sites to catch OWASP Top 10 issues. When a new critical vuln emerges (say a severe Windows bug), agencies often do an emergency scan to identify all systems needing the patch *immediately*. This proactive scanning means you're not waiting for an attacker to find the hole – you find and fix it first.
- **Key Performance Indicators (KPIs):** Agencies measure aspects of their continuous monitoring. For example: mean time to detect an incident (MTTD), percentage of systems sending logs to SIEM, number of high-severity vulnerabilities open beyond SLA, etc. These metrics help identify gaps. If MTTD is high (meaning it takes days to notice breaches), they know they need better detection

tools or more SOC analysts. OMB and DHS have cyber metrics that agencies must report (which often roll up into the FISMA report to Congress). According to GAO, establishing performance measures is an area needing improvement – basically, ensuring continuous monitoring is effective, not just present.

24

Incident Response (IR)

No matter how good defenses are, we must be prepared for incidents. An **incident** could be a malware outbreak, a detected intrusion (e.g., APT in the network), a lost laptop with sensitive data, or even an erroneous release of information. Key elements of IR:

- **Incident Response Plan:** Each agency has an IR plan (often aligning to NIST SP 800-61 Rev2). It spells out roles (incident commander, comms lead, forensics, etc.), categorizes incident severity levels, and provides step-by-step procedures. For example, a "**moderate**" incident might be malware with limited impact, handled by local IT; a "**severe**" might be a major breach requiring a coordinated all-hands approach and reporting to DHS/CISA within 1 hour (as per federal policy for major incidents).
- **Team and Skills:** Agencies maintain or contract **Incident Response Teams** (CSIRT or CIRT). In some cases, CISA's Hunt and Incident Response Team (HIRT) can assist agencies that lack in-house capability, especially for

big incidents. Having skilled forensic analysts is crucial – to analyze compromised systems, they use tools to image memory, collect logs, and trace what happened (and what data might have been exfiltrated). IR teams also need malware analysts to dissect new malware, and they coordinate with threat intel folks to attribute or understand context.

- **Containment, Eradication, Recovery:** These are core stages. **Containment** might involve isolating affected systems (e.g., taking a server offline, blocking an IP range). In a suspected espionage case, there's often a trade-off between containment and monitoring (sometimes agencies let an intrusion persist under watch to gather intel on the adversary, but that's high-stakes and involves NSA/FBI depending on context). **Eradication** means removing the threat – cleaning malware, closing vulnerabilities, changing passwords, etc. **Recovery** is restoring systems to operation (from clean backups, with improved security). For example, after a ransomware on a non-classified network, an agency might wipe and reimage PCs, restore files from backups, patch everything, and only then reconnect them.
- **Communication:** IR plans emphasize communication flows. Internally, informing leadership and affected offices; externally, fulfilling legal reporting requirements. Under FISMA and subsequent OMB guidance, agencies must report certain incidents to Congress, especially if involving personal data (per the Privacy Act and various breach notification laws). Also, CISA must be informed of significant incidents (so they can offer help and share anonymized info with others to prevent spread). Public communication may be necessary if PII was stolen (like

OPM did after their breach). A PR strategy often is part of IR to maintain trust – be transparent but also avoid revealing too much that could help attackers.

- **Lessons Learned:** A critical but sometimes neglected phase is the post-incident analysis. What allowed this incident? How effective was our response? Agencies should convene a "**lessons learned**" meeting to document findings and recommendations. Maybe they discover that while they ultimately kicked out an intruder, it took 6 months to notice – so they invest in better detection. Or that responders struggled to find logs because logging wasn't centralized – so they fix that. Over time, this practice continuously improves the organization's security (turning painful incidents into catalysts for positive change).
- **Testing and Drills:** An IR plan shouldn't sit on a shelf. Regular **tabletop exercises** or even **red team/blue team drills** should be conducted. In a tabletop, you simulate an incident scenario and have the incident handlers verbally walk through what they'd do, which often reveals gaps or confusion to fix. In more advanced drills, a red team might attempt a breach and the blue team has to detect and react in real-time ("cyber war games"). Agencies like to include interagency partners in drills too (e.g., involving CISA or FBI in a practice scenario to make sure coordination works). These drills build muscle memory so that when a real incident hits at 2 AM on a Sunday, the team isn't doing everything for the first time.

25

Continuous Improvement and Adaptation

The threat landscape and technology are ever-changing, so a cybersecurity program must evolve:

- **Emerging Technologies:** Keep abreast of how new tech can both help and harm. For instance, **Artificial Intelligence** can be a double-edged sword: adversaries might use AI to craft better phishing or find vulnerabilities, but defenders can use AI for improved anomaly detection and automated response. Agencies are exploring AI for things like automated risk scoring or predictive analytics to anticipate attacks. However, they also consider the security of AI systems themselves (AI models can be poisoned or tricked).
- **Policy and Framework Updates:** NIST and other bodies continuously update guidelines (we saw SP 800-53 rev 5.2 in 2025, CSF 2.0 coming, etc.). Agencies must adapt their controls to these updates. For example, NIST recently emphasizes

supply chain and IoT security more, so agencies incorporate those controls. When laws update (like if FISMA gets revised by Congress), agencies adjust reporting and programs accordingly. The **National Cybersecurity Strategy Implementation Plan (NCSIP)** released in 2023 outlines many initiatives with deadlines through 2024-2025 that agencies will be implementing – such as shifting liability for insecure software, fostering memory-safe language adoption (some agencies are pushing developers to use Rust or other safe languages for new projects to reduce vulnerabilities).

- **Collaboration and Information Sharing:** No agency stands alone against threats. Continuous improvement involves learning from others. Through groups like the **Cybersecurity and Infrastructure Security Council**, the CIO Council, ISACs (Information Sharing and Analysis Centers for various sectors), agencies share threat intel and best practices. For example, if one agency successfully implemented a particular zero trust solution or thwarted a novel attack, they might publish a case study or talk at a conference so others benefit. Similarly, when one agency gets hit, they often share indicators so others can defend themselves proactively.
- **Metrics and Accountability:** As mentioned, measuring performance is key. OMB's cyber metrics, President's Management Agenda goals, and even public scorecards (the House has occasionally released FISMA scorecards for agencies)

add pressure to improve. Agencies not doing well (say they have low MFA coverage or slow patch times) will face oversight and might have to present improvement plans. This accountability drives continuous improvement efforts at the leadership level.

- **Cultural Change:** Over time, the aim is to integrate security into the DNA of the organization – **security by design** in projects, security considered in every procurement, every employee thinking about security in their daily work (not as a checkbox or afterthought). Many agencies now have **Enterprise Risk Management** boards that include cyber risk alongside other mission risks, elevating it to a strategic discussion. This cultural shift is gradual, but you see it in things like agency heads discussing cybersecurity in strategic plans, not just the CIO or CISO.

In summary, continuous monitoring and incident response are about being vigilant and ready **at all times**, and continuous improvement is about **never being satisfied** – always looking for what can be fortified next. The threat actors are constantly innovating; so must the defenders. A secure government enterprise in 2025 is not the same as in 2015 or 2010 – it's much more dynamic, data-driven, and adaptive, driven by both painful experiences (breaches) and proactive planning. By keeping this cycle of monitor->respond->learn->harden going, agencies aim to stay one step ahead of adversaries and minimize the impact when incidents do occur, thereby protecting the sensitive functions and information entrusted to them.

26

Conclusion

Securing classified, confidential, and sensitive government information is a complex, never-ending journey – but it is absolutely critical to national security and public trust. In this comprehensive exploration, we covered how to engineer and operate a secure environment for government work, touching on every layer from policy to people to technology. Let's recap the key takeaways and why they matter:

- **Information Classification and Handling:** We began with the foundation – understanding the levels of sensitivity (Confidential, Secret, Top Secret) and adhering to the strict rules for each. Without proper classification and need-to-know enforcement, even the best technology can fail (as an insider could access things they shouldn't). The classification system, despite its bureaucracy, provides clarity on what needs protection and to what degree.
- **Governance and Frameworks:** A strong cybersecurity program is guided by robust frameworks (NIST RMF, CSF, 800-53, 800-171/CMMC, etc.) and compliance with laws

and policies (FISMA, EO 14028, OMB mandates, national strategies). These ensure no major aspect of security is overlooked and create accountability. By following these frameworks, agencies implement consistent controls and can measure progress. It also aligns government and industry (through standards like CMMC) so everyone handling government data meets a baseline of security.

- **Zero Trust Architecture:** Trust nothing implicitly – verify everything. This paradigm shift is at the heart of modern government cybersecurity. Implementing zero trust principles (strong identity management, device security, micro-segmentation, continuous authentication) greatly reduces the attack surface and the impact of any single breach. The federal mandate to adopt Zero Trust by 2024 shows how crucial this is. Zero Trust, in essence, is about acknowledging that breaches can happen but designing systems such that a breach in one area doesn't collapse the whole enterprise. It's a bold change from old perimeter models, reflecting today's threat reality.
- **Post-Quantum Cryptography:** Looking to the future, we recognized that today's secrets must remain secure tomorrow. The proactive move to **quantum-resistant encryption** (like CRYSTALS-Kyber and Dilithium) ensures that when quantum computers arrive, they won't unravel years of classified communications. This foresight exemplifies a theme in security – preparing for emerging threats well before they materialize. The government's whole-of-nation effort to migrate to PQC by 2035 underscores that long-term information security (think decades out) is being addressed now, not later.
- **Secure Systems Engineering:** We delved into concrete

practices – hardening systems, segmenting networks, using encryption everywhere, implementing secure configs and continuous patching. These "blocking and tackling" measures, when done thoroughly, eliminate a huge swath of common attack vectors. A securely engineered system means an attacker's job is much harder; even if they get in, they find few easy escalations or juicy targets. It's laborious work, but as often said, **amateurs hack systems, professionals hack people** – by taking care of systems, we force adversaries to resort to social engineering or insider recruitment, which our other measures address.

- **Secure Communications:** We highlighted that protecting data in transit – whether a phone call, email, or data transfer – is just as important as protecting it at rest. Using encryption and secure channels for all government communications (plus training personnel to use them properly and not use personal or insecure channels) closes off avenues of eavesdropping or interception. The image of diplomats using secure phones or agents using one-time pads might come to mind, but equally an everyday civil servant using their PIV card to encrypt an email is part of that same continuum of secure comms.
- **Hardware and Supply Chain:** Supply chain security taught us that we must **trust but verify** everything that goes into our systems, from chips to code libraries. With adversaries possibly attempting to plant backdoors at the factory or in software updates, vigilance at procurement and throughout the lifecycle is essential. The moves to ban risky equipment, require SBOMs, and monitor supplier security all reduce the chance of a "Trojan horse" undermining us from within.

- **Personnel Security and Insider Threats:** Technology cannot defeat an authorized user who turns malicious or careless without help. Thus, investing in people – vetting them with clearances, educating them, monitoring for signs of trouble, and fostering a culture of security – is non-negotiable. The insider threat programs are a testament to hard lessons learned; while you can never have zero risk from insiders, the goal is to deter, detect, and mitigate insider incidents as much as possible. Ultimately, **people are both the strongest and weakest links** – we rely on patriotic, diligent staff to follow procedures and raise concerns, and we guard against the rare few who might stray.
- **Continuous Monitoring and Incident Response:** No defense is perfect, so the ability to quickly **detect and respond** to incidents is crucial. Through SIEMs, SOCs, and IR drills, agencies aim to catch intrusions early (ideally before data is lost) and respond in a coordinated, effective way to minimize damage. Equally, learning from incidents and continuously improving ensures that each scare or breach leads to a stronger posture afterward. The adage "never let a crisis go to waste" applies – every incident is fuel to justify security enhancements that might have been hard to prioritize before.
- **Adapting to Change:** Cybersecurity is a dynamic field. The U.S. government strategies now recognize that things like **AI, cloud, and changing threat actor tactics** require adaptive strategies. The ability to pivot – whether it's updating an outdated system, shifting to a zero trust cloud model, or rapidly deploying a patch enterprisewide – is what distinguishes agencies that will thrive in the face of

cyber aggression. The goal is a nimble security apparatus that can handle everything from a teenage hacker to an elite nation-state group.

Throughout this "book," we cited numerous frameworks, guidelines, and real-world examples to ground these practices in current reality. The sources reinforced that these recommendations are **not merely theoretical**; they are drawn from policy and incidents that have shaped the community's understanding of security. From Biden's quote on bold changes to NIST's finalized PQC standards to GAO's assessment of strategy pillars, it's clear that our approach is informed by the latest knowledge and directives.

As we conclude, remember that securing government operations is **a team effort and a public trust**. It requires commitment at all levels – top leadership must champion and resource it, engineers and analysts must expertly implement it, and every employee must play their part in adhering to it. The stakes could not be higher: the safety of citizens, the integrity of democratic institutions, even lives in some cases, depend on protecting information from adversaries.

In practice, a reader implementing these insights would build a multilayered defense: **classify your data; implement zero trust so that each access is justified; encrypt everything (and prepare for quantum); harden endpoints and networks; vet and train your people; monitor tirelessly; and always be ready to respond.** It's a tall order, but this is what the US government is actively striving for – and significant progress is being made. With each framework adhered to and each new technology securely adopted, we move closer to the ideal of a robust, **resilient**, and **trustworthy** digital

government that can carry out its missions without falling prey to cyber sabotage or espionage.

Finally, cybersecurity is often described as a journey, not a destination. This journey has no finish line, only ongoing improvement. By doing "literally everything" we have outlined – leveraging every framework, tool, and policy at our disposal – we maximize our chances of success. The challenge is immense, but the resolve is strong. In the spirit of continuous improvement, this guide itself would need updates as years go by (perhaps discussing post-quantum cryptography deployment experiences, AI-driven defense, etc.). But the core principles of **diligence, depth, and adaptation** will remain constant. Armed with these, those responsible for securing government operations can face the cyber threats of today and tomorrow with confidence.